C02029: Doctor of Philosophy
CRICOS Code: 009469A
Subject Code: 32903
October 2019

Non-IID Representation Learning on Complex Categorical Data

Chengzhang Zhu

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

Non-IID Representation Learning on Complex Categorical Data

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Analytics

by

Chengzhang Zhu

to

School of Computer Science
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

October 2019

Abstract

Learning complex categorical data requires proper vector or metric representations of the intricate characteristics of that data. Existing methods for categorical data representation usually assume data is independent and identically distributed (IID). However, real-world data is often hierarchically associated with diverse couplings and heterogeneities (i.e., non-IIDness, e.g., various couplings such as value co-occurrences and attribute correlation and dependency, as well as heterogeneities such as heterogeneous distributions or complementary and inconsistent relations). Existing methods either capture only some of these couplings and heterogeneities or simply assume IID data in building their representations.

This thesis aims to deeply understand and effectively represent non-IIDness in categorical data. Specifically, it focuses on (1) modeling heterogeneous couplings within and between attributes in categorical data; (2) disentangling attribute couplings with a mixture of heterogeneous distributions; (3) hierarchically learning heterogeneous couplings; (4) integrating complementary and inconsistent heterogeneous couplings; and (5) adaptively identifying and learning dynamic couplings and heterogeneities.

Accordingly, this thesis proposes (1) a non-IID similarity metrics learning framework to model complex interactions within and between attributes in non-IID categorical data; (2) a decoupled non-IID learning framework to capture and embed heterogeneous distributions in non-IID categorical data with bounded information loss; (3) a heterogeneous metric learning method with hierarchical couplings to learn and integrate the heterogeneous dependencies and distributions in non-IID categorical data into a representation of a similarity metric; (4) an unsupervised heterogeneous coupling learning approach to integrate the complementary and inconsistent heterogeneous couplings in non-IID categorical data; and (5) an unsupervised hierarchical and heterogeneous coupling learning method to learn hierarchical and heterogeneous couplings on dynamic non-IID categorical data.

Theoretical analyses support the effectiveness of the proposed methods and bound the information loss in their generated high-quality representations. Extensive experiments demonstrate that the proposed non-IID representation methods for complex categorical data perform significantly better than state-of-the-art methods in terms of multiple downstream learning tasks and representation-quality evaluation metrics.

AUTHOR'S DECLARATION

I, *Chengzhang Zhu* declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note: Signature removed prior to publication.

Chengzhang Zhu

DATE: 22nd October, 2019

PLACE: Sydney, Australia

Dedication

To my son Jiawei Zhu...

ACKNOWLEDGMENTS

First and foremost I want to thank my supervisor Prof. Longbing Cao. It has been an honor to be his Ph.D. student. He has taught me the skills required by an independent researcher, especially critical thinking and academic writing. I appreciate all his contributions of time, ideas, discussions, and paper polishing to make my Ph.D. experience impressive and stimulating. I am deeply influenced by his professionalism, hard work, and profound ideas. I am also thankful for the excellent opportunity to serve in the research community and the industry internship he has provided.

I would also like to thank Prof. Jianping Yin for his insightful comments, encouragement, and support. His optimism and enthusiasm were contagious and motivational for me, even during the difficult times in my Ph.D. pursuit.

My sincere thanks also go to all of my colleagues in the Advanced Analytics Institute, for their enthusiastic help: Dr. Guansong Pang, Dr. Trong Dinh Thac Do, Dr. Shoujin Wang, Dr. Zhenjiang Lin, Dr. Liang Hu, Qi Zhang, Qing Liu, Zhilin Zhao, Wei Wang, and Jia Xu. I appreciate our intensive discussions, in-depth collaboration, and all the fun we have had in the last four years.

I thank the visiting scholars and students in our laboratory. They broadened the horizon of my research and shared unforgettable times with me at UTS. In particular, I am grateful to Prof. Wentao Zhao, Prof. Lizheng Wang, A/Prof. Quangui Zhang, A/Prof Defu Lian, A/Prof. Lingyun Xiang, Dr. Wenpeng Lu, Dr. Xinwang Liu, Songlei Jian, Shangdong Yang, and Teng Li for their precious advice and encouragement in my Ph.D. research.

I gratefully acknowledge the funding from China Scholarship Council that made my Ph.D. work possible.

Lastly, I would like to thank my family for all their love and encouragement. Without their support, it would not be possible for me to conduct this research. In particular, I am grateful to my wife Qian Li whose spiritual and academic support are so appreciated. Thank you.

Chengzhang Zhu
University of Technology Sydney
October 2019

List of Publications

Related to the Thesis :

Published Journal Papers :

1. **Zhu, C.**, Cao, L., Liu, Q., Yin, J. and Kumar, V., 2018. Heterogeneous metric learning of categorical data with hierarchical couplings. *IEEE Transactions on Knowledge and Data Engineering*, 30(7), pp.1254-1267.
2. Zhao, W., Li, Q., **Zhu, C.**, Song, J., Liu, X. and Yin, J., 2018. Model-aware categorical data embedding: a data-driven approach. *Soft Computing*, 22, pp.3603-3619.

Published Conference Papers :

3. Zhang, Q., Cao, L., **Zhu, C.**, Li, Z. and Sun, J., 2018, July. CoupledCF: Learning explicit and implicit user-item couplings in recommendation for deep collaborative filtering. *In International Joint Conference on Artificial Intelligence* (pp. 3662-3668).
4. Zhao, X., **Zhu, C.** and Cheng, L., 2017, October. Coupled Bayesian matrix factorization in recommender systems. *In 2017 IEEE International Conference on Data Science and Advanced Analytics* (pp. 522-528). IEEE.
5. Song, J., **Zhu, C.**, Zhao, W., Liu, W. and Liu, Q., 2017, September. Model-aware representation learning for categorical data with hierarchical couplings. *In International Conference on Artificial Neural Networks* (pp. 242-249). Springer, Cham.

Submitted Manuscripts :

6. **Zhu, C.**, Cao, L., and Yin, J., 2019. Unsupervised categorical representation with heterogeneous and inconsistent couplings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

7. **Zhu, C.**, Cao, L., Liu, Q., and Yin, J., 2019. Decoupled non-IID nominal data representation. *Journal of Machine Learning Research*.
8. **Zhu, C.**, Cao, L., and Yin, J., 2018. Similarity metrics for non-IID categorical data. *Artificial Intelligence*.
9. Li, Q., Wu, Q., **Zhu, C.**, Cao, L., and Zhang, J., 2019. Deep non-IID paragraph representation for sentiment analysis. *Pattern Recognition*. (Major Revision)

OTHERS :

Published Journal Papers :

10. Yuan, L., Liu, Q., Lu, M., Zhou, S., **Zhu, C**. and Yin, J., 2017. A highly efficient human activity classification method using mobile data from wearable sensors. *International Journal of Sensor Networks*, 25(2), pp.86-92.
11. Zhao, L., Li, K., Wang, M., Yin, J., Zhu, E., Wu, C., Wang, S. and **Zhu, C.**, 2016. Automatic cytoplasm and nuclei segmentation for color cervical smear image using an efficient gap-search MRF. *Computers in biology and medicine*, 71, pp.46-56.
12. Zhou, S., Liu, X., Liu, Q., Wang, S., **Zhu, C.** and Yin, J., 2016. Random Fourier extreme learning machine with $\ell 2$, 1-norm regularization. *Neurocomputing*, 174, pp.143-153.
13. Liu, Q., Zhou, S., **Zhu, C.**, Liu, X. and Yin, J., 2016. MI-ELM: Highly efficient multi-instance learning based on hierarchical extreme learning machine. *Neurocomputing*, 173, pp.1044-1053.

Published Conference Papers :

14. Li, Q., Wu, Q., **Zhu, C.**, and Zhang, J., 2019, September. Bi-level masked multi-scale CNN-RNN networks for short text representation *In International Conference on Document Analysis and Recognition*. IEEE.
15. Li, Q., Wu, Q., **Zhu, C.**, Zhang, J. and Zhao, W., 2019, July. An inferable representation learning for fraud review detection with cold-start problem. *In 2019 International Joint Conference on Neural Networks*. IEEE.
16. Li, Q., Wu, Q., **Zhu, C.**, Zhang, J. and Zhao, W., 2019, April. Unsupervised user behavior representation for fraud review detection with cold-start problem. *In*

Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 222-236). Springer, Cham.

17. Wang, S., Zeng, Y., Liu, Q., **Zhu, C.**, Zhu, E. and Yin, J., 2018, October. Detecting abnormality without knowing normality: A two-stage approach for unsupervised video abnormal event detection. *In 2018 ACM Multimedia Conference on Multimedia Conference* (pp. 636-644). ACM.
18. Zhao, G., Xiang, L., **Zhu, C.** and Li, F., 2018, July. Two-stage unsupervised multiple kernel extreme learning machine. *In 2018 International Joint Conference on Neural Networks*. IEEE.

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Background

Categorical data is pervasive in our daily lives, study, work, entertainment, and social activities. Compared to numerical data analysis, modeling categorical data has been much less intensively explored. The reasons for this lack of study include a lack of effective representation methods to appropriately understand and represent the data complexities intrinsic to categorical data.

In real-life applications, categorical data is not independent; that is, categorical data is associated with diverse *couplings*, for example explicit and implicit value co-occurrences, attribute dependencies, and other interactions between objects, between attributes, and between values (Wu et al. 2004, Cao 2015, Battaglia et al. 2016, Wang, Dong, Zhou, Cao & Chi 2015). Furthermore, categorical data is not identically distributed. It has *heterogeneities* (Zhu et al. 2018); for instance, values in an attribute may be associated with different distributions. The couplings and heterogeneities constitute the non-independent and identically distributed (non-IID) characteristics (Ralaivola et al. 2010, Cao 2014) of categorical data. Complex categorical data with the non-IID characteristics, namely non-IID categorical data, are widely seen in various data applications and learning tasks, such as classification (Zhu et al. 2018), clustering (Jian et al. 2017), and outlier detection (Pang et al. 2016); and representing such data are attracting increasing research interest. Non-IID categorical data is much harder to represent in a numerical space than simple categorical data because of its complicated couplings and

heterogeneities between values, attributes, and objects (i.e., non-IIDness) (Cao 2014).

1.2 Current Work and Gap Analysis

Data representation embeds original data into a vector or similarity/distance space, which significantly affects the performance of downstream learning tasks. Representation is more likely to enable better learning performance if it can comprehensively capture intrinsic data characteristics.

Existing categorical data representation methods usually assume that data is IID. As a result, they may ignore the various couplings and heterogeneities in categorical data. For example, one-hot and dummy encoding use 1 and 0 to indicate the presence or absence of a categorical value, respectively (Powers & Xie 2008). This 1-0 representation treats all categorical values as if they have the same relationship with each other, and thus, it overlooks the complex non-IIDness hidden in categorical data.

Limited efforts have been made to learn the couplings and heterogeneities for categorical data representation (Blei et al. 2003, Ng et al. 2007, Boriah et al. 2008, Cao, Liang, Li, Bai & Dang 2012, Le & Ho 2005, Ahmad & Dey 2007, Ienco et al. 2012, Mikolov et al. 2013, Levy & Goldberg 2014, Jia et al. 2016, Zhang et al. 2015, Peng et al. 2015). For example, Boriah et al. (2008) have introduced a method based on occurrence frequency (OF) to evaluate the similarity between different attribute values, and Blei et al. (2003) have introduced several multinomial distributions with a Dirichlet prior to model heterogeneous value distributions. However, these methods cannot comprehensively represent non-IIDness because they embed either couplings or heterogeneities.

More recently, pioneering work of non-IID learning has been proposed to address the non-IIDness in categorical data, which often exhibits strong couplings and heterogeneities between values, attributes, and objects (Cao et al. 2011, Cao 2014, 2015). The pioneering work has built an initial understanding of the intrinsic nature of the non-IID data and identified how to capture data characteristics. Guided by the non-IID learning framework, hierarchical coupling relationships (Cao, Ou & Yu 2012, Cao 2015) have been considered low-level factors driving attribute and object representation. For example, Wang et al. (2013) and Wang, Dong, Zhou, Cao & Chi (2015) have proposed a similarity representation for categorical data that considers both intra- and inter-attribute couplings and combines them to form coupled object similarity (COS). Subsequently, Jian et al. (2017) have proposed an embedding-based representation method by

hierarchical value coupling learning (CDE). Both COS and CDE perform well capturing complex hierarchical relationships in categorical data. However, they do not jointly consider the heterogeneities in data distributions and dependencies; thus, they fail to distinguish and leverage the redundancy, inconsistency, and complementarity in their captured heterogeneous couplings.

Current research shows that capturing and embedding non-IIDness is critical for categorical data representation. However, current research has not offered an in-depth understanding of non-IIDness. For example, none of the existing methods studies the interaction between heterogeneous distributions and heterogeneous dependencies, and none of them studies hierarchical dynamic couplings and heterogeneities.

Moreover, current research has not built sufficient theoretical foundations for categorical data representation. None of existing methods can theoretically quantify (1) the effectiveness of embedding multiple data characteristics; (2) the information loss in the representation process; and (3) the quality of the representation. Without these theoretical foundations, it is difficult to learn an appropriate categorical representation of satisfying quality.

1.3 Research Problems and Objectives

This research focuses on non-IID representation learning on complex categorical data. It deeply understands and appropriately captures the non-IIDness that is hierarchically embedded in categorical values, attributes, and objects.

This research summarizes the complicated non-IIDness in Figure 1.1. Basically, non-IIDness consists of *couplings*, a term which refers to any interaction between arbitrary entities including values, attributes, objects, and so forth, and *heterogeneities*, including any difference between arbitrary properties such as distributions, relations, formations, and structures. The intersection of couplings and heterogeneities constitutes *heterogeneous couplings*. Heterogeneous couplings, including *heterogeneous dependencies* and *heterogeneous distributions*, are much harder to model than a single coupling or a single heterogeneity. If data contains both heterogeneous dependencies and heterogeneous distributions, the complexity of the data is defined as *non-IID-completeness*.Furthermore, if data has dynamic non-IID-completeness that changes over time, the complexity of the data is defined as *non-IID-hardness*, which indicates the most complex situation in non-IIDness. Obviously, modeling and learning non-IID-completeness and non-IID-hardness on categorical data pose great challenges.

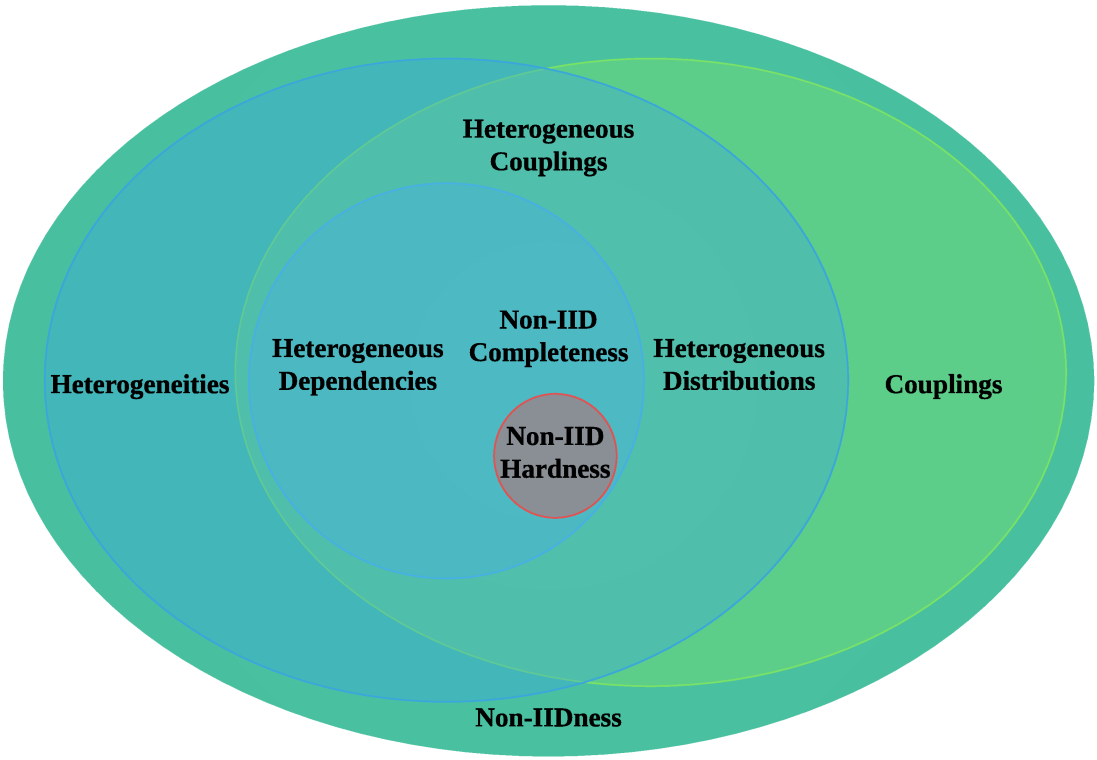


Figure 1.1: Non-IIDness in complex categorical data.

This research investigates how to effectively capture and appropriately represent the non-IIDness in a vector or similarity space on categorical data with different complexities, from simple coupling to non-IID-hardness. Accordingly, this study has the following *research objectives*:

- coupling learning;
- heterogeneity learning;
- non-IID-completeness learning; and
- non-IID-hardness learning.

The *research problems* and their relations to achieving the research objectives have been summarized in Figure 1.2. They include

- couplings learning:
  - modeling couplings within attributes in categorical data;

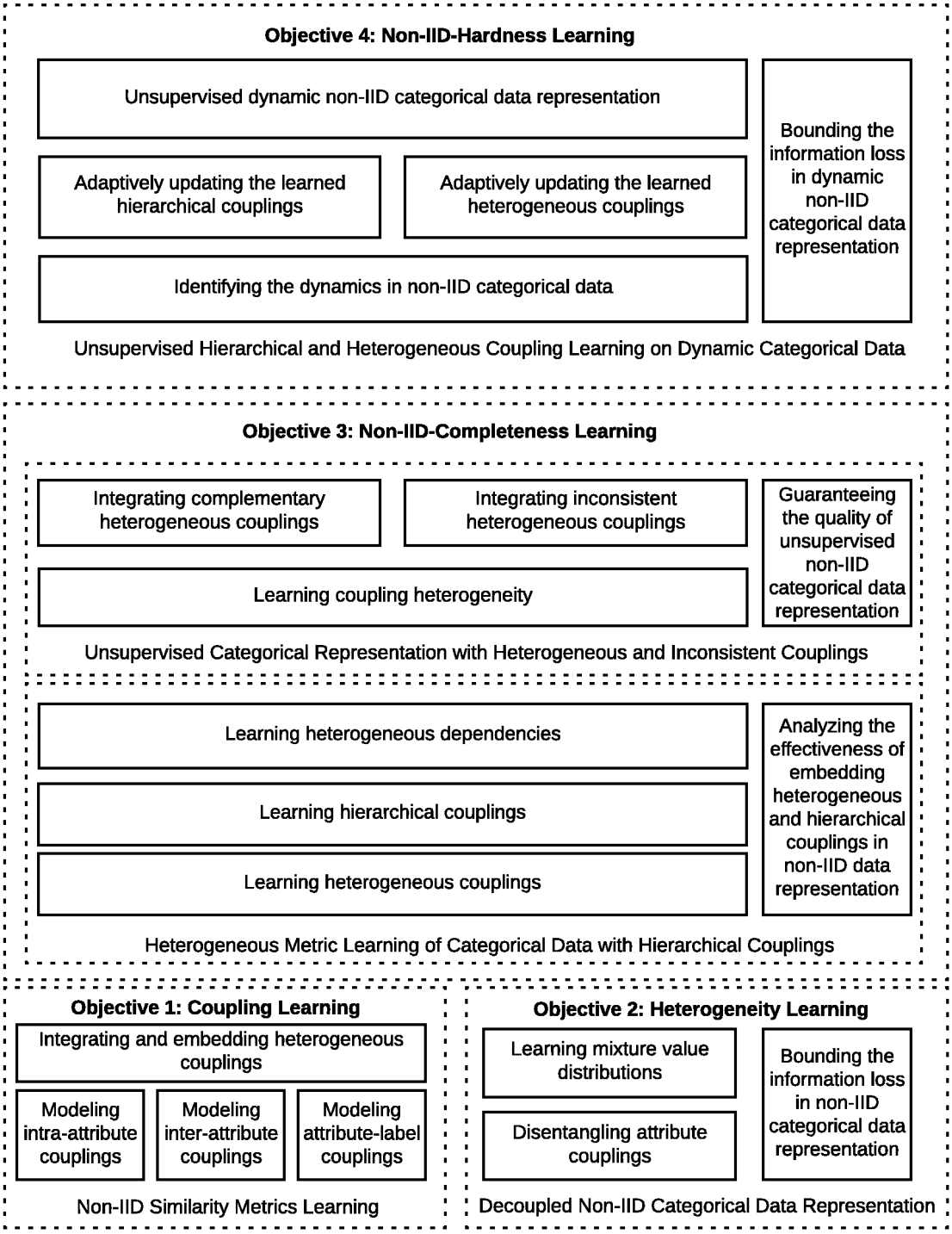


Figure 1.2: The research problems and their relations.

  - modeling couplings between attributes in categorical data;
  - integrating and embedding heterogeneous couplings into a valid metric space;

- heterogeneity learning:
  - learning heterogeneous distributions within an attribute;
  - learning heterogeneous distributions between attributes;
  - disentangling attribute couplings with mixture distributions;
  - bounding the information loss in categorical data representation;

- non-IID-completeness learning:
  - learning heterogeneous and hierarchical couplings in categorical data;
  - learning heterogeneous dependencies in categorical data;
  - analyzing the effectiveness of embedding heterogeneous and hierarchical couplings in categorical data representation;
  - learning coupling heterogeneity;

- non-IID-hardness learning:
  - integrating complementary and inconsistent heterogeneous couplings;
  - identifying the dynamics in non-IID categorical data;
  - adaptively updating the learned hierarchical and heterogeneous couplings; and
  - unsupervised dynamic non-IID categorical data representation.

1.4 Thesis Contributions

This thesis makes the following contributions:

- *Coupling learning:* This thesis studies and models complex interactions within and between attributes in non-IID categorical data to form a similarity metric representation.
- *Heterogeneity learning:* This thesis captures and embeds heterogeneous distributions in non-IID categorical data with bounded information loss.

- *Non-IID-Completeness learning:* This thesis investigates and integrates the heterogeneous dependencies and distributions in non-IID categorical data into a similarity metric representation, and it studies and integrates complementary and inconsistent heterogeneous couplings in non-IID categorical data.
- *Non-IID-Hardness Learning:* This thesis shed light on hierarchical and heterogeneous couplings on dynamic non-IID categorical data.

The above contributions achieve the research objectives and establish a series of non-IID representation learning methods on complex categorical data.

1.5 Thesis Organization

To build a theory and learning system for non-IID representation learning on complex categorical data, this thesis explores four major challenges: coupling learning, heterogeneity learning, non-IID-completeness learning, and non-IID-hardness learning. Accordingly, the thesis consists of substantial work to address each of them: (1) Part I presents coupling learning, as in Chapter 4; (2) Part II presents heterogeneity learning, as in Chapter 5; (3) Part III presents non-IID-completeness learning, including Chapters 6 and 7; and (4) Part IV presents non-IID-hardness learning, as in Chapter 8. The summary of each chapter is as follows:

- *Chapter 2:* This chapter presents a survey of categorical data representation and non-IID learning. Specifically, the categorical data representation paradigms, the coupling learning, heterogeneity learning, non-IID-completeness learning, and non-IID-hardness learning are reviewed and discussed.
- *Chapter 3:* This chapter provides the preliminaries of this research. Specifically, it first presents the symbol styles and key notations used in this thesis. It then defines the basic information functions used for categorical data representation. Finally, it discusses the data sets and primary metrics for representation performance evaluation.
- *Chapter 4:* This chapter studies the research problems of modeling and integrating heterogeneous couplings in complex categorical data to achieve the objective of coupling learning corresponding. Specifically, it proposes a non-IID similarity metrics learning (nSML) framework for categorical data. This nSML framework consists of and integrates multiple similarity metrics to model respective

couplings within and between attributes in non-IID categorical data. The nSML framework is further instantiated by a coupled kernel metric learning (cKML). The metric properties and effectiveness of nSML and cKML will be proved. Extensive experiments on 27 diversified data sets with various data characteristics show that cKML outperforms various existing similarity and distance measures in (1) learning intrinsic similarities and (2) cKML-enabled clustering and classification.

- *Chapter 5:* To achieve heterogeneity learning, this chapter proposes a decoupled non-IID learning (DNL) framework for non-IID categorical data representation. To address the corresponding research problem of disentangling attribute couplings and learning mixture value distributions, the DNL framework first captures attribute couplings and heterogeneity by decomposing non-IID categorical data into IID subspaces, in which attributes are independent of each other and values within an attribute are identically distributed. Then, DNL learns the data characteristics in each IID subspace and the interactions between IID subspaces to represent non-IID categorical data in a Euclidean space. Decoupled non-IID learning is instantiated in a nonparametric Bayesian embedding (non-BEND) model for categorical data representation, built on a Dirichlet multivariate multinomial mixture model. To address the information lost bounding problem, this chapter theoretically analyzes the information loss of the non-IID representation. Theoretical analysis shows that non-BEND approaches zero representation information loss as the amount of data increases. Extensive experiments demonstrate that non-BEND can effectively and efficiently represent large amounts of high-dimensional complex non-IID categorical data, which further significantly improves the clustering, classification, and recommendation performance, as compared with five state-of-the-art categorical representations and one-hot embedding and auto-encoder (AE)-based representations.

- *Chapter 6:* This chapter focuses on the research problem of hierarchical and heterogeneous coupling learning, that of heterogeneous dependencies learning, and that of analyzing couplings-embedding effectiveness to achieve non-IID-completeness learning. It proposes a heterogeneous metric learning with hierarchical couplings (HELIC) for categorical data with hierarchical couplings and heterogeneous distributions. This method captures both low-level value-to-attribute and high-level attribute-to-class hierarchical couplings, and it reveals the intrinsic heterogeneities

embedded in each level of couplings. Theoretical analyses of the effectiveness and generalization error bound will be given to verify that HELIC effectively represents the above complexities. Extensive experiments on 30 data sets with diverse characteristics demonstrate that HELIC significantly enhances the classification accuracy (up to 40.93%), compared with five state-of-the-art baselines.

- *Chapter 7:* This chapter further studies other research problems in non-IID-completeness learning. These research problems include learning coupling heterogeneity, integrating complementary and inconsistent couplings, and the theoretical quality guarantee of unsupervised non-IID categorical data representation. This chapter proposes an unsupervised heterogeneous coupling learning (UNTIE) approach to represent categorical data by both untying the interactions between couplings and revealing the heterogeneous data distributions embedded in each type of couplings. Catering for unsupervised representation, UNTIE is efficiently optimized by seamlessly integrating with a kernel k-means objective function on learning heterogeneous and hierarchical couplings. Theoretical analyses will be given to verify that UNTIE can represent categorical data with maximal separability while effectively representing heterogeneous couplings and their roles in categorical data. Grounded on theoretical analysis, UNTIE can represent categorical data with maximal separability, while effectively represents heterogeneous couplings and their roles in categorical data, enabling significant performance improvement (up to 51.72% F-score improvement) against the state-of-the-art methods on 25 categorical data sets with diversified characteristics.

- *Chapter 8:* To achieve the objective of non-IID-hardness learning, this chapter studies how to identify and learn the coupling dynamics on non-IID categorical data and addresses the research problem of adaptively updating hierarchical and heterogeneous couplings in an unsupervised fashion. Specifically, it proposes an unsupervised hierarchical and heterogeneous coupling learning (UNICORN) method to address the non-IID-hard challenges on dynamic categorical data. This method effectively learns the hierarchical and heterogeneous couplings in categorical data by maximizing the informativeness of UNICORN-generated representations for general learning tasks. To handle dynamic data, UNICORN incrementally enhances the representations of heterogeneous couplings by learning from new observations with the learned heterogeneous couplings as a prior. Comprehensive experiments demonstrate that UNICORN can effectively capture dynamic hetero-

geneous couplings, and its enabled representation quality is significantly better than three state-of-the-art categorical data representation methods in terms of their enabled clustering and object retrieval.

- *Chapter 9:* This chapter summarizes the thesis's content and contributions. It further discusses possible future avenues of research that can build to the work done in this study.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Categorical data representation has a long history with different representation paradigms (e.g., kernel-based representation, distance-based representation, and vector-based representation). Typical categorical data representation methods (e.g., one-hot embedding and Hamming distance) assume categorical data is IID, and thus, they ignore complicated yet essential non-IID characteristics (e.g., couplings and heterogeneities, as discussed in Chapter 1) in categorical data. Advanced categorical data representation methods introduce non-IID learning to capture the complex couplings and heterogeneities. The introduction of non-IID learning significantly enhances categorical data representation performance. However, the existing methods face certain limitations (e.g., overlooking hierarchical couplings; see details in Sections 2.3.4 and 2.4.3) in learning couplings and heterogeneities, and most of them cannot capture non-IID-completeness and non-IID-hardness in complex categorical data.

This chapter presents a survey of categorical data representation and the non-IID learning corresponding. Specifically, it first discusses current categorical data representation paradigms. It then reviews the existing non-IID learning for categorical data representation from four parts corresponding to the thesis' research objectives: (1) coupling learning; (2) heterogeneity learning; (3) non-IID-completeness learning; and (4) non-IID-hardness learning. This survey provides in-depth motivations of the research

problems to achieve the research objectives and points out possible research avenues of non-IID representation learning on complex categorical data.

2.2 Categorical Data Representation Paradigms

2.2.1 Similarity-Based Representation Versus Vector-Based Representation

The work related to categorical data representation can be classed into two paradigms according to whether their embeddings are similarity relevant. The first paradigm is *similarity-based representation*, which represents a categorical data set into a similarity (or dissimilarity) space (Le & Ho 2005, Ng et al. 2007, Ahmad & Dey 2007, Cao, Liang, Li, Bai & Dang 2012, Ienco et al. 2012, Jia et al. 2016, Wang, Dong, Zhou, Cao & Chi 2015, Zhu et al. 2018), and the second paradigm is *vector-based representation*, which represents categorical data in a Euclidean space (Blei et al. 2003, Mikolov et al. 2013, Peng et al. 2015, Zhang et al. 2015, Jian et al. 2017).

In comparison with the similarity-based representation, only limited research has been conducted on the vector-based representation of categorical data. It is much more difficult to represent categorical data into vector space than that into similarity (or dissimilarity) space. This difficulty lies in that similarity (or dissimilarity) space can directly represent the interactions within and between categorical data, but vector space requires many additional operations (e.g., transformation, embedding, and measurement) to persevere these interactions. However, a vector representation of categorical data enjoys the significant benefit of representing a large amount of data with low space complexity. Specifically, for n_o objects, the space complexity of a similarity representation is $O(n_o^2)$ while that of a vector representation is only $O(n_o)$. This thesis first studies similarity-based representation (in Chapter 4) to represent complex couplings in categorical data; then, it studies how to effectively transform a similarity-based representation into a vector-based representation that embeds all these coupling relations to leverage the efficiency of vector-based representation (in Chapters 6, 7 and 8). This thesis also provides a method to generate vector-based representation directly based on non-IID distribution estimation (in Chapter 5).

2.2.2 IID Representation Versus Non-IID Representation

Existing categorical data representation methods can be divided into IID representation and non-IID representation according to their assumptions about categorical data distribution. Most existing methods for categorical data representation belong to IID representation, which assumes categorical data is IID. Under the IID assumption, one easy and efficient way to represent categorical data is to use matching-based methods; for example, the Hamming distance measures the distance between the same values as 0 and between different values as 1. Formally, Hamming distance calculates the distance between two categorical values as follows,

$$d_{HM}(v_i, v_j) = \begin{cases} 0 & \text{if } v_i = v_j \\ 1 & \text{otherwise} \end{cases}, \tag{2.1}$$

where v_i and v_j are values in the same attribute from two objects. Hamming distance can effectively measure the difference between objects by capturing the above attribute value distance. A more powerful alternative to Hamming is *one-hot embedding*, which encodes each categorical value to a numerical vector to represent categorical data by a vector space instead of a dissimilarity space. Denoting the number of unique values in the j-th attribute as $n_v^{(j)}$, the numerical vector of the j-th attribute will have $n_v^{(j)}$ dimensions, each of which corresponds to a categorical value, and for the i-th categorical value, the i-th dimension will be 1 and others be 0. Different from the Hamming distance, which simply matches attribute values to form a binary similarity, Akusok et al. (2014) have proposed a method that adopts an extended Jaccard distance to measure the similarity between attribute values. Specifically, the method divides attributes into different groups according to their different contributions. It then calculates the similarity between two objects, o_i and o_j, as follows:

$$s_{akusok}(o_i, o_j) = \frac{1}{|K|} \sum_{k \in K} \frac{|A_i^{(k)} \cap A_j^{(k)}|}{|A_i^{(k)}| + |A_j^{(k)}| - |A_i^{(k)} \cap A_j^{(k)}|}, \tag{2.2}$$

where $A_i^{(k)}$ and $A_j^{(k)}$ are the k-th attribute sets for object o_i and o_j, respectively, and $K = K_i \cap K_j$ and K_i (respectively K_j) is the set of attribute indexes for object o_i. Accordingly, this method embeds attribute value distributions into a categorical data similarity-based representation, assuming that different values in an attribute are independent. Another advanced IID representation method is the heterogeneous support vector machine (SVM) proposed by Peng et al. (2015). Instead of simply matching, this method

learns a categorical data representation that can maximize the SVM's theoretical margin.

However, the IID representation methods treat each categorical value equally; thus, they cannot appropriately understand and capture the rich characteristics (e.g., couplings and heterogeneities) in categorical data. Let us illustrate the issues relevant to the above considerations through a toy example in Table 2.1. Categorical data in this example cannot be simply described by matching-based methods, although these methods are the most straightforward and common ways to measure categorical data similarity. When a matching-based measure (e.g., Hamming distance or matching kernel; Gärtner et al., 2004) is used, the similarities between color *yellow* and *green* and between *yellow* and *black* are both 0; however, *yellow* is closer to *green* than *yellow* is to *black* from the semantic perspective.

Table 2.1: Toy Example: The Watermelon Information Table. Each Watermelon with a Different Sweetness is Described with respect to Three Attributes: *Texture*, *Color*, and *Root Shape*.

ID	**Texture**	**Color**	**Root Shape**	**Sweetness**
A1	clear	white	straight	low
A2	blurry	yellow	straight	low
A3	blurry	yellow	curled	low
A4	clear	green	slightly curled	low
A5	blurry	green	curled	high
A6	clear	black	slightly curled	high

The challenges in complex categorical data representation arise from intrinsic non-IID data complexities, including the diverse couplings and heterogeneities. On the one hand, attribute couplings contain much richer information than the usually concerned relations, such as value co-occurrences and marginal attribute distributions, going beyond the IID assumption of attributes and objects. The rich relations between attributes indicate that non-IID data distributions cannot be modeled by the product of distributions of individual attributes. On the other hand, heterogeneities involve the difference of mixed distributions in each attribute and between attributes, which makes the joint distribution in an attribute much harder to estimate, compared with homogeneous attributes which share the same distributions. With the above characteristics, directly modeling non-IID categorical data with n_o objects and n_a attributes would require $n_v^{(1)} \times n_v^{(2)} \times \cdots \times n_v^{(n_a)}$ parameters, where $n_v^{(j)}$ is the number of unique values of the j-th attribute. In contrast, the number of parameters required in the IID case is only

$n_v^{(1)} + n_v^{(2)} + \cdots + n_v^{(n_a)}$. As a result, effectively representing complex categorical data requires non-IID representation, which has bigger challenges compared with IID representation.

Although the existing non-IID representation methods introduce non-IID learning methods to enhance categorical data representation (see details in Sections 2.3–2.5), they lack solid theoretical support for (1) the effectiveness of embedding multiple data characteristics; (2) the information lost in the representation process; and (3) the quality of the representation. Without these theoretical foundations, it is hard to guide appropriate learning with a satisfying representation quality. To build these theoretical foundations, this thesis introduces a serial of theoretical tools to prove the effectiveness of embedding multiple data characteristics (in Chapter 6), to bound the information lost in the representation process (in Chapter 5), and to quantify representation quality (in Chapters 6 and 7).

2.3 Coupling Learning

The quality of categorical data representations are affected by how well a representation captures the various value-to-object coupling relationships (Cao 2015, Ng et al. 2007, Boriah et al. 2008). Furthermore, categorical data representations can be further enhanced by learning more types of couplings (Cao 2015), as evidenced by (1) intra-attribute coupling-based representations (Ng et al. 2007, Cao, Liang, Li, Bai & Dang 2012) and (2) inter-attribute coupling-based representations (Le & Ho 2005, Ahmad & Dey 2007, Boriah et al. 2008, Jia et al. 2016). On the one hand, the *intra-attribute coupling-based representations* reveal the way and degree that values are coupled within an attribute. For example, the method proposed by Ng et al. (2007) adopts the conditional probability of the attribute values of an object with respect to the attribute cluster centers to represent categorical data, and the method proposed by Cao, Liang, Li, Bai & Dang (2012) introduces set theory to measure intra-attribute value similarity to represent categorical data. On the other hand, the *inter-attribute coupling-based representations* aim to capture the way and extent that attributes are coupled. They typically measure the inter-attribute couplings with respect to the conditional probabilities (Le & Ho 2005, Ahmad & Dey 2007) or co-occurrence frequencies (Jia et al. 2016) between values of different attributes. These two groups of representations outperform other classical methods, such as matching-based, as they capture richer interactions in categorical data. In addition, leveraging attribute-label couplings also shows promising

for categorical data representation (Cheng et al. 2004, Xie et al. 2012, Cheung & Jia 2013, Shi et al. n.d., Zhu et al. 2018), especially for a certain downstream learning task.

Theoretical progress made in coupling learning has then been integrated into categorical data clustering (Wang, Dong, Zhou, Cao & Chi 2015, Ienco et al. 2012, Qian et al. 2016), classification (Zhu et al. 2018), regression (Tutz & Gertheiss 2016), ensemble learning (Krawczyk et al. 2017, Wang et al. 2018), and applications including outlier detection (Pang et al. 2016) and recommendation (Zhang et al. 2018). These avenues of research represent new opportunities and significant performance improvements over existing learning designs and representation methods.

2.3.1 Learning Intra-Attribute Couplings

In this subsection, intra-attribute coupling learning methods for categorical data representation are discussed. These methods represent categorical values by the couplings within an attribute.

To leverage the impacts of all values in an attribute, Ng et al. (2007) proposed a method that represents categorical data dissimilarity by considering the conditional probability of each categorical value given different object clusters, which reflects relations between all values. Denoting an object cluster as C, the mode of the j-th attribute in this cluster as $m^{(j)}$, the value in the j-th attribute of the i-th object as $o_i^{(j)}$, this method calculates the attribute-level dissimilarity between the cluster C and the object o_i as follows:

$$d_{Ng}(C, o_i) = \begin{cases} 1 & \text{if } o_i^{(j)} \neq m^{(j)} \\ 1 - \frac{|\{o_k | o_k^{(j)} = o_i^{(j)}, o_k \in O_c\}|}{|O_c|} & \text{otherwise} \end{cases}, \tag{2.3}$$

where O_c refers to the set of objects in the cluster C and where $|\cdot|$ represents the number of objects in a set. The method further calculates the object-level dissimilarity by a weighted summation of the attribute-level dissimilarity per each attribute. However, Ng et al. (2007) did not discuss how to determine appropriate summation weights. To adaptively assign the weights, Cheung & Jia (2013) proposed a method to calculate the weight of each attribute according to the attribute entropy. Compared with the previous methods, the above two methods capture contextual information in each attribute; thus, they achieve better representation performance. However, these methods have two essential drawbacks because they have to iteratively update object similarity and

clusters. The first drawback is that the errors in the clustering process may be propagated and magnified to the representation process. As a result, these methods are very sensitive to the adopted clustering method. The second disadvantage is the large cost of time caused by the similarity updating in each iteration. Consequently, these methods are computationally expensive for complex data sets that require several clustering iterations to capture the data complexities.

A variety of efficient methods to capture the relations of all values in an attribute is proposed by Boriah et al. (2008). A representative method among these is *occurrence frequency* (OF), which gives higher similarity for mismatches on more frequent values and lower similarity for that on less frequent values. Given a set of objects, it calculates the similarity between two values, v_i and v_j, as follows:

$$s_{OF}(v_i, v_j) = \begin{cases} 1 & \text{if } v_i = v_j \\ \\ \frac{1}{1+\log\frac{n_o}{|O_{v_i}|}\cdot\log\frac{n_o}{|O_{v_j}|}} & \text{otherwise} \end{cases}, \tag{2.4}$$

where n_o is the number of objects, $|\cdot|$ refers to the size of a set, and O_{v_i} and O_{v_j} refer to the set of objects with value v_i and v_j, respectively. Compared with the method proposed by Ng et al. (2007), which uses clusters to reflect relations between all values, OF uses the attribute value distribution to describe the value relations. However, this method implicitly assumes the occurrence of categorical values in each attribute has a concave probability density function, an assumption which does not hold in most real cases, since the occurrence of most categorical values follows normal distribution that has a convex probability density function. Furthermore, OF ignores the couplings between attributes. As a result, OF is suited only to measuring the data with independent attributes where the occurrence of values appears as a concave distribution.

In addition to probability theory, set theory is also introduced to capture intra-attribute couplings. For example, Cao, Liang, Li, Bai & Dang (2012) proposed a set theory-based method that measures the similarity between attribute values by a rough membership function. Given two attribute values, v_i and v_j, this method calculates their similarity as follows,

$$s_{rough}(v_i, v_j) = \frac{v_i \equiv v_j}{|O_{v_i}|}, \tag{2.5}$$

where O_{v_i} refers to the set of objects with value v_i and

$$v_i \equiv v_j = \begin{cases} 1 & \text{if } v_i = v_j \\ \\ 0 & \text{otherwise} \end{cases}. \tag{2.6}$$

As can be seen from the formula, this method captures the universal distribution of values in each attribute as the intra-attribute couplings. However, the captured couplings provide detailed information only of each categorical value, ignoring differences between different categorical value pairs.

On the contrary, Wang et al. (2011) introduced an intra-attribute coupling learning method from the frequency perspective. For attribute values v_i and v_j, this method learns their intra-coupling by

$$s_{Ia} = \frac{|O_{v_i}| \cdot |O_{v_j}|}{|O_{v_i}| + |O_{v_j}| + |O_{v_i}| \cdot |O_{v_j}|}, \tag{2.7}$$

where $|O_{v_i}|$ and $|O_{v_j}|$ are the number of objects with values v_i and v_j, respectively. Compared to the work of Cao, Liang, Li, Bai & Dang (2012), Wang et al.'s (2011) work not only considers the distribution of one value but also captures the joint distribution of value pairs. Therefore, it can discover more useful information, especially for pair-wise similarity calculation. However, the drawbacks of this method include that (1) the frequency-based method cannot distinguish values with different meanings but with same frequency; (2) this method requires a strong assumption that the occurrence of values in an attribute has a convex probability dense function, whereas most real occurrences of attribute values obey a concave probability function; and (3) this representation does not validate metric properties, which are required for many learning algorithms.

2.3.2 Learning Inter-Attribute Couplings

Inter-attribute coupling learning methods represent categorical data by embedding the attribute dependencies. Specifically, they represent the similarity between categorical values by considering their couplings with values in other attributes. In this section, recent work focusing on the inter-attribute coupling learning will be analyzed.

To capture the inter-attribute relations, Le & Ho (2005) first proposed a method to measure the similarity of categorical attribute values depending on their relationships with other attributes. Specifically, it calculates the dissimilarity between two values in

an attribute by two steps. In the first step, it estimates the probabilities of a value in another attribute conditioned on these two values, respectively. In a second step, it measures the Kullack–Leibler divergence (Kullback & Leibler 1951) between the estimated conditional probabilities to form the dissimilarity between the two values. Formally, given two values, v_i and v_j, in the m-th attribute, this method calculates their dissimilarity as follows:

$$d_{Le}(v_i, v_j) = \sum_{v_k \in V^{(l)}, l \neq m} (p(v_k|v_i) \log \frac{p(v_k|v_i)}{p(v_k|v_j)} + p(v_k|v_j) \log \frac{p(v_k|v_j)}{p(v_k|v_i)}), \tag{2.8}$$

where V_j refers to the set of values of j-th attribute, and

$$p(v_k|v_i) = \frac{|O_{v_k}|}{|O_{v_i}|}. \tag{2.9}$$

Instead of estimating conditional probabilities of all values in other attributes, the method proposed by Ahmad & Dey (2007) calculates the conditional probability of a set of values in each attribute. Specifically, for two values in an attribute, it firstly searches a set of values in another attribute that can maximize the probability of the set conditioned on the first value and the probability of the complement set conditioned on the second value. It then adopts a procedure similar to that of Le & Ho (2005) to calculate the dissimilarity between the two values, v_i and v_j, in the m-th attribute as follows:

$$s_{Ahmad}(v_i, v_j) = \frac{1}{n_a - 1} (\sum_{l=1, l \neq m}^{n_a} \max_{W_l} (p_m^{v_i}(W_l) + p_m^{v_j}(\sim W_l) - 1)), \tag{2.10}$$

where W_l is a subset of values in the l-th attribute, $\sim W_l$ is the complement value set of W_l in the l-th attribute, and $p_m^{v_i}(W_l)$ is the probability of W_l conditioned on value v_i in the m-th attribute.

Further, Wang et al. (2011) have proposed a novel measure to capture inter-attribute couplings. This measure has been implemented by universal set, joint set, and intersection-set-based methods. Since these three implementations have been proved with the same effect, the intersection set-based method, which has the least computational cost, is selected to illustrate the intuition of the measure proposed by Wang et al. (2011). Formally, the intersection set-based inter-attribute similarity is defined as follows:

$$s_{Ie}(v_i, v_j) = \sum_{l=1, l \neq m}^{n_a} \alpha_l \sum_{v_k \in V^{(l)}} \min\{p(v_k|v_i), p(v_k|v_j)\}, \tag{2.11}$$

where the probability function $p(\cdot|\cdot)$ is the same as defined in Equation (2.9).

Although the above methods introduce inter-attribute couplings in categorical data representation, they assume all attributes have the same affection for an object. Obviously, this strong assumption may not hold in most real cases, which further induces an inaccurate representation.

Recently, a new distance metric for unsupervised categorical data representation has been proposed by Jia et al. (2016). This method uses the inter-attribute values' probability of occurring to measure two values' distance, offering a more solid theoretical explanation. Moreover, this method not only selects suitable attributes for object-similarity calculation but also considers how to combine each value similarity to form the object similarity. Specifically, it emphasizes the value similarity in the attribute with low variance and vice versa. Therefore, it fills the gap that remains in previous work (Le & Ho 2005, Ahmad & Dey 2007, Ienco et al. 2012). For two values, $v_i^{(m)}$ and $v_j^{(m)}$, in the m-th attribute, this method calculates its similarity by:

(2.12)
$$d_{Jia}(v_i^{(m)}, v_j^{(m)}) = \begin{cases} \sum_{l \in S^{(m)}} r(m,l) p((v_i^{(m)}, v_i^{(l)}) = (v_j^{(m)}, v_j^{(l)})) & \text{if } v_i^{(m)} \neq v_j^{(m)} \\ \\ \sum_{l \in S^{(m)}} r(m,l) \delta(v_i^{(l)}, v_j^{(l)}) p((v_i^{(m)}, v_i^{(l)}) = (v_j^{(m)}, v_j^{(l)})) & \text{if } v_i^{(m)} = v_j^{(m)} \end{cases},$$

where

(2.13)
$$r(m,l) = \frac{I(V^{(m)}; V^{(l)})}{H(V^{(m)}, V^{(l)})},$$

$I(V^{(m)}; V^{(l)}))$ and $H(V^{(m)}, V^{(l)})$ are the mutual information and joint entropy of value sets $V^{(m)}$ and $V^{(l)}$, respectively. Forthermore,

(2.14)
$$s(i) = \{a_l | r(m,l) > \beta, 1 \leq l \leq n_a\},$$

(2.15)
$$\delta(v_i^{(m)}, v_j^{(m)}) = \begin{cases} 1 & \text{if } v_i^{(m)} \neq v_j^{(m)} \\ \\ 0 & \text{if } v_i^{(m)} = v_j^{(m)} \end{cases},$$

and

(2.16)
$$p((v_i^{(m)}, v_i^{(l)}) = (v_j^{(m)}, v_i^{(l)})) = p(v^{(m)} = v_i^{(m)} \wedge v^{(l)} = v_i^{(l)}) \cdot p^-(v^{(m)} = v_i^{(m)} \wedge v^{(l)} = v_i^{(l)})$$

(2.17)
$$+ p(v^{(m)} = v_j^{(m)} \wedge v^{(l)} = v_j^{(l)}) \cdot p^-(v^{(m)} = v_j^{(m)} \wedge v^{(l)} = v_j^{(l)}),$$

where $v^{(m)} \in V^{(m)}$ and $v^{(l)} \in V^{(l)}$.

In Equation (2.16), $p(\cdot)$ calculates the proportion of a subset in the whole set, and $p^{-}(\cdot)$ has a definition similar to that of $p^{(\cdot)}$, yet removes an object with a given attribute value in the population.

2.3.3 Learning Attribute-Label Couplings

Many methods embed attribute-label couplings for data representation. Typically, they learn a transformation vector or matrix to map data from the original space to a specific space where the data distribution is more suitable for a certain learning task. Most of these methods focus on numerical data (Weinberger & Saul 2009, Ying & Li 2012, Lim et al. 2013, Lim & Lanckriet 2014, Cuturi & Avis 2014, Liu et al. 2015, Ye et al. n.d.), and few of them study categorical input (Cheng et al. 2004, Xie et al. 2013, Cheung & Jia 2013, Shi et al. n.d., Zhu et al. 2018).

The method proposed by Stanfill & Waltz (1986) first considers attribute-label couplings for categorical data representation. It uses the label to divide data into subsets and considers the attribute value distribution within these subsets. The difference of appearance frequencies of attribute values is used as a distance measure, called value difference metric (VDM). For attribute values v_i and v_j, the VDM is defined as

$$d_{VDM}(v_i, v_j) = \sum_{c \in C} (p(c|v_i) - p(c|v_j))^2, \tag{2.18}$$

where C is the set of all class labels and where function $p(\cdot|\cdot)$ is defined as in Equation (2.9). A significant merit of this method is the calculated similarity's suitability for classification. However, considering only the distribution in label-induced subsets may cause an overfitting problem. More information, such as intra- and inter-attribute couplings, should be added to modify this metric.

Cheng et al. (2004) have revised the method proposed by Stanfill & Waltz (1986) by considering inter-attribute couplings and wrapping the similarity measure in a learning process. The similarity learned by this method has a higher performance for classification, as compared to previous work. However, this method is time-consuming. A trade-off is needed between accuracy and consumed time.

Instead of calculating the metric directly, Jain et al. (2012) study the connection between metric learning and kernel learning, and they transform a metric learning problem into a kernel learning task. Through kernel learning, the similarity of different types of data, including categorical data, can be measured. Although this method can

be used for various types of data, it ignores the data characteristics of these data and has high space complexity.

Another work that adopts the kernel method to measure distance is proposed by He et al. (2013). This work proposes a so-called kernel density metric learning (KDML) method, which provides a non-linear and probability-based similarity measure. This method first maps data into a feature space, based on a kernel density estimation. Then, existing metric learning methods are used in the newly constructed feature space to determine the Mahalanobis distance. This idea has a remarkable advantage in that it can handle heterogeneous data, such as data that mixes numerical and categorical values. However, KDML suffers the weakness that it uses only a matching method to capture the relationship in original data. This method will lead to information loss in the new feature space. Thus, more low-level information should be captured to build a fundamental similarity. Another shortage of KDML is that it considers only a certain kernel to map original data. However, different kernels are suited for different data distributions. A uniform kernel cannot be used to handle all cases appropriately. How to select a suitable kernel to fit data characteristic must be considered in further details.

In place of simply embedding attribute-label couplings, Peng et al. (2015) propose a method that represents categorical data by jointly considering attribute-label couplings and a downstream learning model. Specifically, this method searches for data representation that can maximize the generalization error bound of the downstream learning model, using the attribute-label-coupling embedded vector as the initial search point. However, this method only maps a categorical value to a numerical value. As a result, it could not well represent a categorical value if the value contained rich information that should be embedded in a high-dimensional space.

2.3.4 Discussion

The aforementioned coupling learning methods show superior performance over different data characteristics. However, they also have some clear weakness in categorical data representation.

Intra-attribute coupling learning methods discover low-level characteristics of attribute values (e.g. their individual and joint distribution). However, they ignore the interactions between attributes (i.e. the contextual information). Although the inter-attribute coupling learning methods reveal this contextual information, most of them ignore low-level relationships between attribute values and simply treat these values as same or not the same. The more recent work on coupled object similarity (COS) cap-

tures both value and attribute interactions, but the proposed similarity measures are not metric-based and do not provide a solid theoretical foundation for metric operations and wide applicability. To tackle these problems, this thesis proposes a novel non-IID similarity metrics learning framework to capture and embed comprehensive and hierarchical couplings into a valid metric space (see details in Chapter 4).

The attribute-label coupling learning can generate a representation that is more suited to a specific downstream learning task compared with intra- and inter-attribute coupling learning. However, attribute-label coupling learning methods should further consider how to integrate with intra- and inter-attribute couplings and how to avoid overfitting problems. This thesis studies these problems in Chapter 6 and proposes a novel heterogeneous metric learning method, which hierarchically captures intra- and inter-attribute and attribute-label couplings, with bounded generalization errors.

Furthermore, most of these coupling learning methods ignore heterogeneities in complex categorical data. Consequently, they treat the distributions for value, attribute, and object in categorical data as homogeneous, and they usually adopt one measure for all data but ignore the differences between values (and attributes and objects) and their relations. However, effectively capturing heterogeneities is required by complex categorical data representation (see details in Sections 2.4 and 2.5). This thesis thus studies heterogeneity learning in Chapter 5 and studies how to effectively integrate couplings and heterogeneities in Chapters 6 and 7.

2.4 Heterogeneity Learning

Heterogeneity is an important non-IIDness characteristic of complex categorical data. Generally, it includes heterogeneous distributions and heterogeneous dependencies at the value-to-attribute-to-object level. Capturing and embedding the heterogeneity is critical for categorical data representation, which may significantly improve downstream learning performance (Zhang et al. 2015).

2.4.1 Learning Heterogeneous Distributions

Heterogeneous distribution learning methods target discovering and representing different distributions in categorical data. For example, Blei et al. (2003) have proposed a latent Dirichlet analysis (LDA) method to estimate multiple distributions in categorical data. Specifically, the LDA method introduces several multinomial distributions with

a Dirichlet prior to model heterogeneous categorical value distributions, and it represents a categorical value by its estimated occurrence probabilities in these multinomial distributions. For natural language processing (NLP) tasks, LDA-generated representation achieves remarkable learning performance. However, LDA requires a prior about the number of distributions, which means it cannot fully recognize the heterogeneity in categorical data. Furthermore, LDA is designed for a single variable, so it cannot conduct on categorical data with multiple attributes. These problems also appear in other categorical data representation methods in NLP area, such as that proposed by Mikolov et al. (2013) and Levy & Goldberg (2014). On the contrary, a series of methods (Lazarsfeld et al. 1968, Kolda 2001, Dunson & Xing 2009) aims to decompose mixture distributions in categorical data into multiple independent multinational distributions. However, these methods do not embed heterogeneous distributions into categorical data representation. Recently, Zhang et al. (2015) proposed a multiple-distance-based representation method to handle various distributions in multivariate categorical data, but it assumes attributes are independent and ignores the hierarchical interactions in categorical data.

Although heterogeneous distribution learning is researched only by limited categorical data representation methods, it has been widely studied by many related areas in machine learning, such as in multiple kernel learning (Gönen & Alpaydın 2011, Wang et al. 2017, Liu et al. 2019), multiple view learning (Kan et al. 2015, Xu et al. 2015, Wang, Arora, Livescu & Bilmes 2015), and mixture distribution modeling (Blei et al. 2006, Xue et al. 2007, Rodriguez et al. 2008). In these areas, many heterogeneous distribution learning methods have been proposed and have demonstrated advances. However, most of these methods are designed for numerical data, so they cannot handle categorical data directly or be simply converted to learn heterogeneous distributions in categorical data because (1) they usually estimate distribution with respect to numerical vectors with certain intervals, which violates the nature of categorical data, and (2) transforming categorical data to an appropriate numerical representation is nontrivial and may change the original distributions.

2.4.2 Learning Heterogeneous Dependencies

Heterogeneous dependency learning methods focus on capturing and integrating multiple types of couplings at the same or different levels.

To capture the heterogeneity in inter-attribute couplings, Ienco et al. (2012) introduced symmetric uncertainty (SU) as a criteria to measure the correlation of attributes.

Formally, SU between attributes a_j and a_k is defined as follows:

$$SU_{a_j}(a_k) = 2 \times \frac{H(a_j) - H(a_j|a_k)}{H(a_j) + H(a_k)}, \tag{2.19}$$

where $H(a_j)$ and $H(a_j|a_k)$ are the entropy and condition entropy of variables a_j and a_k, respectively. Based on SU, Ienco et al. (2012) designed a rule to determine the set of attributes (namely context) that have strong couplings but are slightly redundant with an attribute under SU measure. For an attribute a_j, the rule first puts all other attributes in its context. It then iteratively removes an attribute a_k from the context if the following conditions are satisfied per another attribute a_m in that context:

$$(1) : SU_{a_j}(a_k) \leq SU_{a_j}(a_m) \tag{2.20}$$

and

$$(2) : SU_{a_m}(a_k) \geq SU_{a_j}(a_k). \tag{2.21}$$

After determining the context of an attribute, this method calculates the co-occurrence probability of two values in the attribute in terms of all values in its contextual attributes to form the representation of these two values. Specifically, this method represents the dissimilarity of two categorical values, v_i and v_j, as follows:

$$d_{Ienco}(v_i, v_j) = \sqrt{\frac{\sum_{k \in context(i)} \sum_{v \in V^{(k)}} (P(v|v_i) - P(v|v_j))^2}{\sum_{k \in context(i)} |V^{(k)}|}}, \tag{2.22}$$

where $context(i)$ refers to the context of the i-th attribute under SU measure. Compared with the categorical data representation method proposed by Ahmad & Dey (2007), the method proposed by Ienco et al. (2012) selectively captures heterogeneous dependencies that enhance the representation's performance. Ienco & Pensa (2016) further combine these heterogeneous inter-attribute couplings into a one-class learning method for categorical data representation, achieving significant performance improvement.

Following the introduction of SU, Khorshidpour et al. (2011) introduced two more context selection methods in their work, including a method based on mutual prediction and a method based on the Chi-square test, to learn the heterogeneity in the couplings captured by Le & Ho (2005). For an attribute, these context selection methods can appropriately find contextual attributes that have a strong correlation with the attribute. As demonstrated by the experimental results, leveraging this heterogeneity dramatically improves the performance of categorical data representation.

All of the aforementioned methods consider the heterogeneity in heterogeneous inter-attribute couplings, but they ignore the heterogeneity between inter-attribute couplings and intra-attribute couplings. Wang et al. (2011) have proposed a method that integrates heterogeneous intra- and inter-attribute couplings to construct categorical data representation. Wang, Dong, Zhou, Cao & Chi (2015) further adopt SU to select contextual attributes for each attribute when integrating inter-attribute couplings. However, these methods use multiplication to integrate intra- and inter-couplings, which leaves only consensus information and ignores complementary information in these couplings.

Although the above methods recognize that inter-attribute couplings do not exist between all attributes, they fail to appropriately integrate the learned heterogeneous inter-attribute couplings. Specifically, they represent object-level couplings by equally summing all inter-attribute couplings. However, different inter-attribute couplings may make different contributions to the object-level coupling because they may not be independent and may have different relations to downstream learning tasks.

Recently, Jian et al. (2017) proposed an embedding-based representation for categorical data by hierarchical value coupling learning (CDE), which successfully models various coupling relationships between categorical values and embeds them as categorical data representations. To integrate the multiple couplings, CDE reduces their redundancy and retains their important information by principal component analysis (PCA) method. However, it treats attributes as independent, which may not hold for non-IID categorical data. In addition, it ignores the inconsistency, which reflects diverse interactions and distributions, between different couplings. In addition, this method has quadratic time complexity in terms of the number of attributes and the number of categorical values. As a result, it may not be able to represent very high-dimensional complex categorical data.

2.4.3 Discussion

In summary, the existing heterogeneous distribution learning methods either cannot conduct on general categorical data (e.g., the representation methods in NLP area work only for individual categorical variables) or overlook the couplings between attributes (e.g., the multiple-distance-based representation assumes attributes are independent). Consequently, these methods may fail to model the heterogeneous distributions in complex categorical data. This thesis studies heterogeneous distribution learning for categorical data in Chapter 5, where a decoupled non-IID learning framework is proposed to disentangle attribute couplings and learn mixture value distributions.

The current heterogeneous dependency learning methods achieve significant performance especially in reducing redundancy of multiple couplings. However, while they reveal the diverse interactions and relations in different types of couplings, they ignore that couplings may be both complementary and inconsistent with each other, namely coupling heterogeneity, as illustrated by the example in Table 2.1. If an intra-attribute coupling (i.e., value couplings) is measured in terms of value frequency, the difference between slightly curled and curled watermelons of the attribute *root shape* is 0, due to their same frequency. However, the difference between slightly curled and curled watermelons is not 0 because the curled root is more related to the yellow and green watermelons while the slightly curled root is more associated with the green and black ones, considering the inter-attribute couplings between *color* and *root shape*. The inconsistency between couplings is one kind of heterogeneities (Ralaivola et al. 2010, Cao 2014). This inconsistency may be essentially caused by (1) different types of couplings corresponding to different interactions in data that may follow different data distributions and (2) different data distributions possibly existing in a data set. To mitigate the effect of inconsistency, this thesis systematically studies how to effectively capture such coupling heterogeneity (see details in Chapter 7).

2.5 Non-IID-Completeness and Non-IID-Hardness Learning

2.5.1 Learning Non-IID-Completeness

Most of the current categorical data representation methods only capture and embed either couplings or heterogeneity (Stanfill & Waltz 1986, Blei et al. 2003, Le & Ho 2005, Ahmad & Dey 2007, Boriah et al. 2008, Cao, Liang, Li, Bai & Dang 2012, Jia et al. 2016). Although limited efforts have been made concerning learning heterogeneous couplings for categorical data representation (Ienco et al. 2012, Khorshidpour et al. 2011, Wang, Dong, Zhou, Cao & Chi 2015, Zhang et al. 2015, Jian et al. 2017), theses studies do not jointly capture the heterogeneous dependencies and heterogeneous distributions in non-IID-complete categorical data. As a result, they may introduce biases in the representation because they embed complicated data complexities from only incomplete aspects. To comprehensively represent categorical data, non-IID-completeness learning is required to leverage both heterogeneous dependencies and heterogeneous distributions and appropriately integrate them by considering their interactions. To achieve this

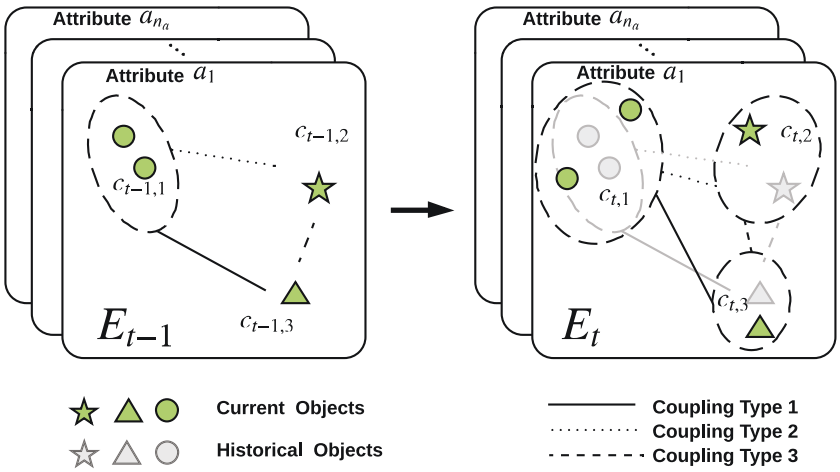


Figure 2.1: Hierarchical and heterogeneous couplings in dynamic categorical data.

goal, this thesis studies supervised and unsupervised non-IID-completeness learning for categorical data representation in Chapters 6 and 7, respectively.

2.5.2 Learning Non-IID-Hardness

Non-IID-hard categorical data consists of non-IID-completeness that changes over time. In addition to static heterogeneous dependencies and distributions in values, attributes, and objects, non-IID-hard categorical data contain hierarchical dynamic couplings and distributions. Consequently, non-IID-hardness learning is required to model the non-IID-completeness at each time point and to capture the coupling relations between non-IID-completeness over time.

Figure 2.1 illustrates the non-IID-hardness in dynamic categorical data. First, as shown in the left part of the diagram, there may exist different grouping effects (which can be captured by similarity) in objects (e.g., c_1, c_2 and c_3) with respect to different categorical attributes; such groupings may change from one categorical attribute to another. For example, student researchers working in a research lab may consist of undergraduate students (e.g., c_1), master's students (c_2), and doctoral students (c_3), who may share certain similarities (e.g., with respect to the attributes research area and cultural background) while also keep their heterogeneous characteristics (e.g., with respect to research method). These different couplings between various attributes are associated with the couplings between students with respect to their qualifications. Further, when discussing the student similarities with respect to another categorical attribute (e.g.,

religious background), these students can be grouped in other ways (e.g., as Christian students, Muslim students, Buddhist students, and nonbelievers etc.). Moreover, the couplings within and between these groups differ from those in the above qualification-based groups. This simple example illustrates that heterogeneous couplings may exist between objects of the same or different groups, as reflected and coupled in terms of different categorical attributes. Second, non-IID-hardness learning particularly concerns the dynamic categorical data widely seen in real-life applications (i.e., as shown in the right part of Figure 2.1, new objects may appear at time t in each coupled group of objects at time $t-1$). Accordingly, the couplings within and between categorical attributes also change from time $t-1$ to t, and the same is true of the couplings between objects. The above hierarchical, heterogeneous, and dynamic couplings between objects and between attributes are widely seen in the real world (e.g., between different types of investors trading in a capital market or in an online shopping marketplace).

Handling the above challenges forms the main motivation and objective of non-IID-hardness learning. This objective requires identifying and representing multiple types of couplings and heterogeneities (i.e., non-IID-completeness), which reflect diverse interactions and distributions of objects, at each time point (e.g., $t-1$ in Figure 2.1) and to capture the connections between objects and the variation of non-IID-completeness between objects at different time points (e.g., $t-1$ and t in Figure 2.1).

Although various methods have been proposed to analyze dynamic categorical data for different applications, such as stream learning (Cao et al. 2010), clustering change detection (Chen et al. 2009), and concept drift detection (Bai et al. 2016), none of these methods captures non-IID-hardness. For example, methods for clustering change detection in categorical data—including information entropy-based clustering (Barbará et al. 2002, Li et al. 2014), importance-based clustering (Chen et al. 2009, Bai et al. 2016), and fuzzy theory-based clustering (Cao et al. 2010, Saha & Maulik 2014)—ignore the hierarchical and heterogeneous couplings, object changes, and their coupling changes at different time points. In addition, deep recurrent models (Bengio et al. 2013) can abstract latent relations and sequential relations, but these models ignore observable relations and their incremental changes over time. To learn the non-IID-hardness on complex categorical data, this thesis further studies how to adaptively update heterogeneous and hierarchical couplings to form dynamic non-IID categorical data representation based on non-IID-completeness learning (see details in Chapter 8).

CHAPTER 3

PRELIMINARIES

3.1 Introduction

This chapter first presents the symbol styles and key notations used in this thesis. It then defines basic information functions for categorical data representation. After that, it discusses the categorical data sets used in this thesis. It finally introduces the primary performance-evaluation metrics for categorical data representation.

3.2 Symbol Styles and Key Notations

This thesis uses the symbol styles listed in Table 3.1. Specifically, it uses lowercase with sans serif font to represent *element*, uses lowercase to represent *value*, uses lower case with bold font to represent *vector*, uses uppercase with bold font to represent *matrix*, uses uppercase to represent *set*, uses lowercase with parentheses to represent *function*, uses uppercase with Calligraphic font to represent *space*, uses subscript for *value index*, and use superscript with parenthesis for *attribute index*.

Following the symbol styles, this thesis presents a static categorical data set as a three-element tuple $E =< O, A, V >$, where $O = \{\mathsf{o}_i | i \in N_o\}$ is an object set with n_o objects; $A = \{\mathsf{a}_i | i \in N_a\}$ is an attribute set with n_a attributes; and $V = \bigcup_{j=1}^{n_a} V^{(j)}$ is the collection of attribute values with n_v values, in which $V^{(j)} = \{\mathsf{v}_i^{(j)} | i \in N_v^{(j)}\}$ is the set of attribute values $\mathsf{v}_i^{(j)}$ with $n_v^{(j)}$ values of attribute a_j, and N_o, N_a and $N_v^{(j)}$ are the sets of indices

Table 3.1: List of Symbol Styles

Symbol	**Symbol Style**
Element	Lowercase with sans serif font
Value	Lowercase
Vector	Lowercase with bold font
Matrix	Uppercase with bold font
Set	Uppercase
Function	Lowercase with parentheses
Space	Uppercase with calligraphic font
Value index	Subscript
Attribute index	Superscript with parenthesis

for objects, attributes, and values of the j-th attribute, respectively. For the i-th object o_i, the categorical value in the j-th attribute a_j can be represented as $v_i^{(j)}$. For example, in Table 2.1, O = {A1, A2, A3, A4, A5, A6}, A = {Texture, Color, Root Shape}, V = {clear, blurry, white, yellow, green, black, straight, curled, slightly curled}, $V^{(1)}$ = {clear, blurry}, and $v_2^{(3)}$ = straight.

Furthermore, this thesis presents a dynamic categorical data set with n_t time points as a sequential matrix of three-element tuples $E = \{E_1, \cdots, E_t, \cdots, E_{n_t}\}$, where $E_t =< O_t, A, V_t >$ refers to a three-element tuple of categorical data observed at time t, where $O_t = \{o_{t,i} | i = 1, \cdots, n_{o_t}\}$ is an object set with n_{o_t} objects, where $A = \{a_j | j = 1, \cdots, n_a\}$ is an attribute set with n_a attributes, and where $V_t = \bigcup_{j=1}^{n_a} V_t^{(j)}$ is a collection of attribute values with n_{v_t} values, in which $V_t^{(j)} = \{v_{t,i}^{(j)} | i = 1, \cdots, n_{v_t}^{(j)}\}$ is a set with $n_{v_t}^{(j)}$ values of attribute a_j. In the case of dynamic data notation, a subscript denotes a time point. For example, O_t refers to the observed object set at time t. The index of an element is also denoted by a subscript followed by the time point notation with a comma to split them. For example, $o_{t,i}$ refers to the i-th object in O_t at time t. The index of an attribute is denoted as a superscript with a parentheses. For example, $V_t^{(j)}$ refers to the set of values of j-th attribute at time t. When there is no ambiguity, this chapter omits the time point and index for a more concise representation. For example, at time t, $v^{(j)}$ is used to represent a categorical value in the j-th attribute. To represent the output of a model for dynamic categorical data, this thesis uses a subscript to denote the time point of the input data and a superscript to denote the time point at which the model is learned or needs to be learned. For example, $\mathbf{x}_t^{t-1}$ refers to the numerical representation of an object observed at time t generated by the model learned at time $t-1$, and $\mathscr{M}_t^t$ refers to the coupling spaces of observations at time t learned by coupling

learning functions at time t.

Other notations used in this thesis are listed in Section A.1 and Section A.2 for static and dynamic data, respectively.

3.3 Basic Information Functions

Let the relationships between objects, attributes, and attribute values be represented by a set of functions $\{v^{(j)}(\cdot)|j \in N_a\}$, in which $v^{(j)}(\cdot) : O \rightarrow V^{(j)}$ maps an object to a particular value with respect to attribute a_j. Let the relationships between objects and classes be represented by the function $c(\cdot) : O \rightarrow C$, which maps an object to its corresponding class or classes. For example, in Table 2.1, a relationship between object, attribute, and value for A2 is $v^{(2)}(\text{A2}) = yellow$, and a relationship between object and class for A2 is $c(\text{A2}) = low$. With the above functions, the following basic information functions are defined to learn the hierarchical couplings in the proposed methods.

Definition 3.1 (Attribute-to-object mapping function [AOF])**.** AOF, denoted as $g^{(j)}(\cdot)$, is a mapping function that maps the values in a value set of the j-th attribute to their corresponding objects in the data set:

$$g^{(j)}(V_*^{(j)}) = \{\mathsf{o}_i | v^{(j)}(\mathsf{o}_i) \in V_*^{(j)}\}, \tag{3.1}$$

where $V_*^{(j)} \subseteq V^{(j)}$ is a subset of the attribute $\mathsf{a}^{(j)}$'s value set. Accordingly, $g^{(j)}(V_*^{(j)})$ returns those objects having the given values in $V_*^{(j)}$.

For example, in Table 2.1, $g^{(3)}(\{slightly\ curled\})$ = {A4, A6} and $g^{(2)}(\{white, black\})$ = {A1, A6}.

Definition 3.2 (Class-to-object mapping function [COF])**.** Denoted as $h(\cdot)$, COF is a mapping function that finds those objects whose class labels are contained in a given class set:

$$h(C_*) = \{\mathsf{o}_i | c(\mathsf{o}_i) \in C_*\}, \tag{3.2}$$

where $C_* \subseteq C$ is a subset of all classes.

For example, $h(\{low\})$ = {A1, A2, A3, A4} and $h(\{high\})$ = {A5, A6}, as shown in Table 2.1.

Definition 3.3 (Value-frequency-percentage function [VFF])**.** The VFF, denoted as $f_j(v_i^{(j)})$, is a function that calculates the percentage of a specific value $v_i^{(j)}$ in the attribute a_j corresponding to the object o_i in terms of the value frequency. The VFF is formalized as

$$f_j(v_i^{(j)}) = \frac{|g_j(v_i^{(j)})|}{n_o}.$$

For example, $f_2(v_1^{(j)}) = f_2(white) = \frac{1}{6}$ and $f_3(curled) = \frac{1}{3}$ in Table 2.1.

Definition 3.4 (Information conditional probability functions [ICPFs])**.** Information conditional probability functions are to calculate the information conditional probability of the value set of a categorical attribute with respect to the set of another attribute, following Bayes' theorem. Here, ICPF is denoted as $p_{j|k}(\cdot|\cdot)$ to calculate two sets of values from different attributes, and as $p_{j|c}(\cdot|\cdot)$ to calculate a set of values and a set of classes.

Given value subset $V_*^{(j)}$ of attribute a_j and value subset $V_*^{(k)}$ of attribute a_k, ICPF $p_{j|k}(V_*^{(j)}|V_*^{(k)})$ is calculated as follows:

$$p_{j|k}(V_*^{(j)}|V_*^{(k)}) = \frac{|g^{(j)}(V_*^{(j)}) \cap g^{(k)}(V_*^{(k)})|}{|g^{(k)}(V_*^{(k)})|}. \tag{3.3}$$

Given value subset $V_*^{(j)}$ of attribute a_j and class subset C_*, the ICPF $p_{j|c}(V_*^{(j)}|C_*)$ is calculated as

$$p_{j|c}(V_*^{(j)}|C_*) = \frac{|g^{(j)}(V_*^{(j)}) \cap h(C_*)|}{|h(C_*)|}. \tag{3.4}$$

In Equations (3.3) and (3.4), $\cap$ calculates the intersection of two sets, and $|\cdot|$ returns the number of elements in a given set.

For example, in Table 2.1, the ICPF of the value $curled$ of attribute *root shape* with respect to the value *yellow* of attribute *color* is

$$p_{3|2}(\{curled\}|\{yellow\}) = \frac{|\{\mathrm{A3},\mathrm{A5}\} \cap \{\mathrm{A2},\mathrm{A3}\}|}{|\{\mathrm{A2},\mathrm{A3}\}|} = \frac{1}{2},$$

and with respect to the class *low* is

$$\begin{aligned} & p_{3|c}(\{curled\}|\{low\}) \\ &= \frac{|\{\mathrm{A3},\mathrm{A5}\} \cap \{\mathrm{A1},\mathrm{A2},\mathrm{A3},\mathrm{A4}\}|}{|\{\mathrm{A1},\mathrm{A2},\mathrm{A3},\mathrm{A4}\}|} \\ &= \frac{|\{\mathrm{A3}\}|}{|\{\mathrm{A1},\mathrm{A2},\mathrm{A3},\mathrm{A4}\}|} = \frac{1}{4}. \end{aligned}$$

For simplicity, the following parts of this thesis use $p(\cdot|\cdot)$ to represent the ICPF for both cases (i.e., between two sets of values and between a set of values and a set of classes) when doing so causes no confusion.

3.4 Data Sets for Representation Performance Evaluation

To evaluate the categorical data representation performance, this thesis uses 32 categorical data sets[1] from different areas in its experiments. These comprise medical data: Lymphography (Lym), Hepatitis (Hep), Audiology (Aud), Primarytumor (Prim), Spect (Spc), Wisconsin(Wcs), BreastCancer (Br), Dermatology (Dmg), and Mammographic (Ma); gene data: DNAPromoter (DNAP), DNANominal (DNAN) and Splice (Spc); social and census data: HouseVotes (Hsv), Adult (Adt), Census (Cens) and Hayesroth (Hay); hierarchical decision-making data: Monks3 (Monk), Krvskp (Krv), Tictactoe (Tic), Krkopt (Krk) and Connect4 (Cnt); nature data: SoybeanSmall (SoyS), SoybeanLarge (SoyL), Zoo, Flare (Flr) and Mushroom (Ms); Business data: Crx; psychological experimental data: Balance (Ba); disaster prediction data: Titanic (Titn); and synthetic data: Mofn3710 (Mof), ThreeOf9 (Tr) and Led24 (Ld).

These 32 data sets have strong diversity in terms of data factors: the number of objects (n_o), the number of attributes (n_a), the number of classes (n_c), the average number of attribute values (n_{av}), the maximal number of attribute values (n_{mv}), and the class imbalance (CI) rate (r_{CI}). The r_{CI} (see Definition 3.5 below) is measured by the ratio between the largest number of objects in a class and the smallest number of objects in another class.

Table 3.2 summarizes the data characteristics. It shows that the number of objects ranges from 101 to 299,285, and the number of attributes ranges from 3 to 69. The data sets contain both binary and multiple classes with the maximum number of 24 classes. The average and maximal numbers of attribute values range from 2 to 16.09 and from 2 to 53, respectively. The imbalance rate ranges from 1 to 168.63. These diverse characteristics are used to test the sensitivity of each representation method on diverse data characteristics.

Definition 3.5. *The CI rate*:

$$r_{CI} = \max_{i,j} b(c_i, c_j),$$

where $b(c_i, c_j)$ is defined as follows, and i and j refer to different class labels c_i and c_j.

$$b(c_i, c_j) = \begin{cases} \frac{|g_c(c_i)|}{|g_c(c_j)|} & \text{if } |g_c(c_i)| > |g_c(c_j)| \\ 1 & \text{otherwise} \end{cases}.$$

[1]They are downloaded from the following websites: http://archive.ics.uci.edu/ml; https://www.sgi.com/tech/mlc/db; and https://www.kaggle.com.

Table 3.2: Characteristics of Benchmark Data Sets

Data set	**Abbr.**	n_o	n_a	n_c	n_{av}	n_{mv}	r_{CI}
SoybeanSmall	SoyS	47	35	4	2.06	7	1.7
Zoo	Zoo	101	16	7	2.25	6	10.25
DNAPromoter	DNAP	106	57	2	4.0	4	1.0
Hayesroth	Hay	132	4	3	3.75	4	1.7
Lymphography	Lym	148	18	4	3.28	8	40.5
Hepatitis	Hep	155	13	2	2.77	3	3.84
Audiology	Aud	200	69	24	2.23	7	48.0
HouseVotes	Hsv	232	16	2	2.0	2	1.15
Spect	Spc	267	22	2	2.0	2	3.85
Mofn3710	Mof	300	10	2	2.0	2	4.0
SoybeanLarge	SoyL	307	35	19	3.77	8	40.0
Primarytumor	Prim	339	17	21	2.47	4	84.0
Dermatology	Dmg	366	33	6	3.91	4	5.6
Monks3	Monk	432	6	2	2.83	4	1.12
ThreeOf9	Tr	512	9	2	2.0	2	1.15
Balance	Ba	625	4	3	5.0	5	5.88
Wisconsin	Wcs	683	9	2	9.89	10	1.86
Crx	Crx	690	9	2	5.0	15	1.25
BreastCancer	Br	699	9	2	10.0	11	1.9
Mammographic	Ma	830	4	2	5.0	7	1.06
Tictactoe	Tic	958	9	2	3.0	3	1.89
Flare	Flr	1,066	11	6	3.73	8	7.7
Titanic	Titn	2,201	3	4	2.0	2	3.11
DNANominal	DNAN	3,186	60	3	4.0	4	2.16
Splice	Spl	3,190	60	3	4.78	6	2.16
Krvskp	Krv	3,196	36	2	2.03	3	1.09
Led24	Ld	3,200	24	10	2.0	2	1.14
Mushroom	Ms	5,644	22	2	4.45	9	1.62
Krkopt	Krk	28,056	6	18	6.67	8	168.63
Adult	Adt	30,162	8	2	12.25	41	3.02
Connect4	Cnt	67,557	42	3	3.0	3	6.9
Census	Cens	299,285	35	2	16.09	53	15.12

For the experiments of supervised representation learning (e.g., in Section 6.4) and downstream supervised learning tasks (e.g., in Sections 4.4.1.2, 4.4.2.1, and 5.6.2 etc.), Monte Carlo cross-validation is taken to partition a data set to training and test sets. Compared with other validation methods, Monte Carlo cross-validation can largely retain the heterogeneous distributions in the training set. Hence, it is more suitable for

evaluating the representation of complex categorical data, which may have strong heterogeneous distributions. Specifically, the experiments randomly select 90% of objects in each data set for training and use the remainder for testing, and 20 random sampling iterations generate 20 sets of training and test data for the experiments.

3.5 Metrics for Representation Performance Evaluation

To empirically evaluate categorical data representation performance, this thesis introduces two kinds of evaluation methods: *representation-quality-based evaluation* and *down-stream-task-based evaluation*. While the former evaluates representation performance from the representation itself, the latter tests representation performance by the performance of downstream learning tasks (e.g., classification and clustering).

3.5.1 Representation-Quality-Based Evaluation Metrics

A typical representation-quality-based evaluation method is visualization. Specifically, the representation vectors or representation similarity (or dissimilarity) matrix generated by a method is visualized in a two-dimensional space; some visible properties (e.g., separability and structurality) are used to quantitatively evaluate the representation performance. In order to visualize representation into a two-dimensional space, this thesis introduces t-distributed stochastic neighbor embedding (t-SNE) (Maaten & Hinton 2008) and multidimensional scaling (MDS) (Borg & Groenen 2005) to transform high-dimensional vectors and similarity matrix into two-dimensional space, respectively. The basic idea of t-SNE is to minimize the Kullback-Leibler divergence between the data distribution in the original high-dimensional space and that in the transformed low-dimensional space. Accordingly, the low-dimensional visualization can reflect the main distribution characteristics of the high-dimensional representation. The MDS method transforms objects from similarity space to a low-dimensional Cartesian space where the between-object similarities are preserved as well as possible.

Another representation-quality-based evaluation metric used in this thesis is the (ϵ, γ)-good criterion proposed by Balcan et al. (2008). Following Definition 4 in Balcan et al. (2008), a similarity function $s(\cdot, \cdot)$ is strongly (ϵ, γ)-good for a learning problem $\mathscr{P}$

if at least a $1-\epsilon$ probability mass of objects o satisfies

$$\begin{aligned} &E_{\mathsf{o},\mathsf{o}'\sim\Phi}[s(\mathsf{o},\mathsf{o}')|c(\mathsf{o})=c(\mathsf{o}')] \\ &\geqslant E_{\mathsf{o},\mathsf{o}'\sim\Phi}[s(\mathsf{o},\mathsf{o}')|c(\mathsf{o})\neq c(\mathsf{o}')]+\gamma, \end{aligned}$$

where E refers to the expected value on distribution Φ in the learning problem $\mathscr{P}$ and where $c(\mathsf{o})$ refers to the class of o. In this criterion, ϵ indicates the proportion of objects whose averaged intra-class similarity is not γ degrees larger than their averaged inter-class similarity value. With the same γ, the smaller ϵ reflects the better similarity function for the learning problem $\mathscr{P}$. Here, γ indicates the extent to which the intra-class similarity is larger than the inter-class similarity on the $1-\epsilon$ proportion of data that is best separated. When ϵ is fixed, the larger γ reflects the better similarity function. The intuition of this criterion is that a good similarity measure should effectively differentiate data in the same class from those in other classes. More importantly, a (ϵ,γ)-good similarity can induce a classifier with a bounded error (see more details in Theorem 1 from Balcan et al., 2008). Since different ϵ values may correspond to different γ values, in experiment, the (ϵ,γ)-curves are drawn to demonstrate the quality of the representation learned by each method. With the same ϵ, the better representation would have a greater γ. In other words, the better representation would yield a higher curve in the (ϵ,γ)-curve figure.

3.5.2 Down-Stream-Task-Based Evaluation Metrics

This thesis feeds categorical representation to several downstream tasks, including classification, clustering, object retrieval, and recommender system, to form down-stream-task-based evaluation methods.

For classification and clustering tasks, to avoid subjective judgment and CI, two metrics, F-score and normalized mutual information (NMI), may be introduced to evaluate the learning performance in the experiment. A higher F-score or a higher NMI indicates better learning accuracy. This thesis adopts macro-averaged F-score, which is popular in class imbalanced settings (Narasimhan et al. 2016), to measure the gap between the classification (or clustering) results and ground truth. Formally, the macro-averaged F-score can be calculated as follows,

$$F\text{-score} = \frac{1}{n_c}\sum_{i=1}^{n_c}\frac{2\times\text{Precision}_i\times\text{Recall}_i}{\text{Precision}_i+\text{Recall}_i}, \tag{3.5}$$

where Precision_i refers to the precision with respect to the i-th class (or cluster) c_i (i.e., the fraction of objects classified or clustered as belonging to c_i that in fact do belong to

c_i) and where Recall_i refers to the recall with respect to c_i (i.e., the fraction of objects of c_i that are correctly classified or clustered). Denoting the ground-truth class (or cluster) set as C and the estimated set $\hat{C}$, the NMI metric is calculates as follows,

$$\text{NMI}(C,\hat{C}) = \frac{2 \times I(C;\hat{C})}{H(C)+H(\hat{C})}, \tag{3.6}$$

where $I(C;\hat{C})$ refers to the mutual information between C and $\hat{C}$ and where $H(C)$ and $H(\hat{C})$ refer to the entropy of C and $\hat{C}$, respectively. To fairly compare different representation methods on multiple data sets, this thesis evaluates the averaged rank (AR) of each method over all data sets.

For object retrieval task, this thesis uses every object as a query and retrieves its k-closest objects per their representation. The precision@k (i.e., the fraction of the retrieved k objects are the same-class neighbors) and recall@k (i.e., the fraction of the same-class neighbors retrieved in the k objects) are reported as the evaluation metrics, which demonstrate the quality of learned representation from local (when k is small) to global (when k is large).

Part I

Coupling Learning

CHAPTER 4

NON-IID SIMILARITY METRICS LEARNING ON CATEGORICAL DATA

4.1 Introduction

Genuinely quantifying the similarities in categorical data is a fundamental task for categorical data analysis. As discussed in Section 2.2.1, similarity-based representation can directly reflect the couplings within and between categorical data, and much attention has been paid to similarity-based coupling learning.

However, most existing coupling learning methods captures only specific characteristics of categorical data (see details in Section 2.3): (1) the matching-based measures do not distinguish value ranges and frequencies; (2) many methods measure object similarity without involving within- and between-attribute relationships; (3) the existing value-based measures capture only specific data characteristics (e.g., either the relationships within one dimension or between two attributes without involving value interactions); and (4) the more recent work (e.g., COS; Wang, Dong, Zhou, Cao & Chi, 2015) captures both value and attribute interactions, but the proposed similarity measures are not metric-based and do not provide a solid theoretical foundation for metric operations and wide applicability.

To address the above issues in coupling learning, this chapter proposes a non-IID similarity metrics learning (nSML) framework for modeling and integrating hetero-

geneous couplings in complex categorical data. This framework transfers the original non-IID feature space to a multi-metric space that captures comprehensive and hierarchical non-IID categorical data relations. In particular, nSML captures the following data characteristics and complexities in categorical data:

- The rich coupling relationships (couplings, for short) between the values of an attribute, called *intra-attribute couplings*, measure the within-attribute similarities; the intra-attribute similarities reflect the value distribution and interactions within an attribute.
- The rich couplings between attributes, called *inter-attribute couplings* or *attribute interactions*, measure the between-attribute similarities; the inter-attribute similarities indicate value interactions in an attribute, conditional on other attributes.
- The rich couplings between objects combine both intra- and inter-attribute similarities. This combination aggregates the value-to-attribute-to-object similarities.

A further goal of the present work is to instantiate the nSML framework in terms of a coupled kernel metric learning (cKML) approach, which learns the hierarchical heterogeneous couplings and their aggregation in non-IID categorical data in terms of different kernels. This method (1) learns the above value-to-attribute-to-object similarities by taking a data-driven approach to capture the underlying intrinsic data characteristics in the low-level non-IID attribute space; (2) represents the above heterogeneous aspects of similarities in terms of diversified kernels; (3) integrates the respective similarities in a high-level kernel space; and (4) induces a dissimilarity metric for non-IID objects in a transformed space.

The key contributions of this chapter include the following:

- *A general framework:* It captures comprehensive non-IID characteristics and aspects in non-IID categorical data for in-depth similarity learning.
- *Hierarchical similarity metrics:* Several kernel-based similarity metrics are proposed to measure hierarchical and heterogeneous non-IID aspects in categorical data. Specifically, the hierarchical coupling relationships in data are learned by modeling the value-to-object couplings in terms of value frequency and co-occurrences within and between categorical attributes and between the object interactions. These low-level hierarchical value-to-object couplings are incorporated into different kernel functions to measure object couplings and to complement and enhance the high-level kernel model performance.

- *A valid object-distance metric:* This study further proves that the proposed measures are positive semi-definite kernels and the induced distance measures hold metric properties. Hence, they can be used to integrate different similarity information and are sufficiently flexible to handle various kernel-based tasks.
- *A comprehensive metric-effectiveness evaluation:* The proposed cKML instantiates the nSML framework and is fed in several methods for learning categorical data. This learning method is compared with different mechanism-based similarity measures for the clustering and classification of 27 data sets with different data characteristics. The results show that cKML significantly improves the learning performance in data sets with sophisticated relations.

The rest of this chapter is organized as follows: Section 4.2 gives the details of the proposed nSML and its instantiation model cKML. Section 4.3 presents the theoretical analysis of the cKML properties. Section 4.4 demonstrates the performance of cKML by comparing it with existing categorical similarity measures in a variety of learning tasks. Section 4.5 concludes this chapter and discusses prospects for future research.

4.2 The Non-IID Similarity Metrics Learning Framework

This section proposes a non-IID similarity metrics learning framework for categorical data. As shown in Figure 4.1, the nSML framework presents as a bottom-up hierarchical structure that introduces several measures to capture intra-attribute coupling (k_{Ia}) and inter-attribute coupling (k_{Ie}) from heterogeneous categorical data. This framework further integrates k_{Ia} and k_{Ie} into object coupling (k_O). Finally, a non-IID distance metric ($d(\cdot,\cdot)$) is obtained by applying a transformation function to the object coupling.

The attribute coupling measures capture the coupling relationships within and between attributes. The intra-attribute coupling measure ($k_{Ia}^{(j)}(\cdot,\cdot)$) captures the attribute value interactions according to the value distribution of the attribute a_j. The inter-attribute coupling measure ($k_{Ie}^{(j,k)}(\cdot,\cdot)$), on the other hand, captures the attribute value relations in terms of the value interactions between the values of a_j and of a_k. Lastly, a non-IID object coupling measure ($k_O(\cdot,\cdot)$) measures the similarity between two objects. It integrates the coupling relationships between two objects by accumulating attribute couplings. Based on the non-IID object-coupling measure, a non-IID distance metric ($d(\cdot,\cdot)$) can be induced through a transformation function.

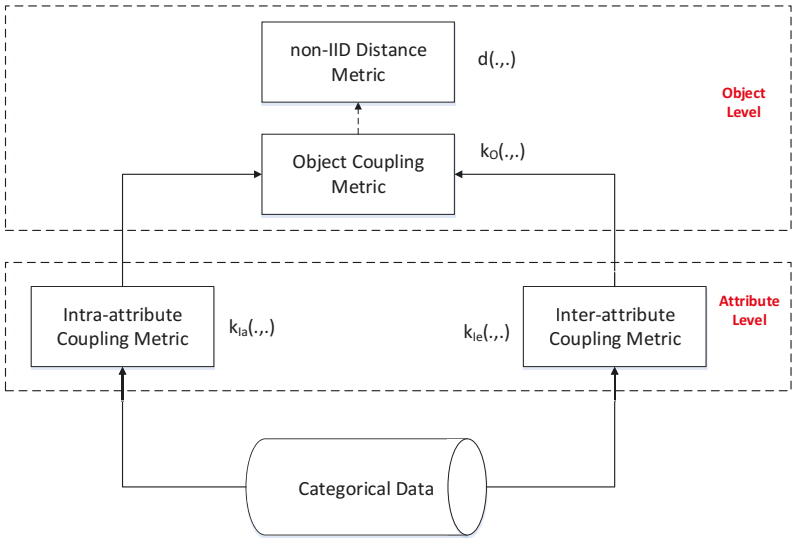


Figure 4.1: The non-IID similarity metrics learning framework

4.2.1 Coupled Kernel Metric for Categorical Data

This section instantiates the nSML framework in terms of a cKML system. It is worth noting that the proposed nSML framework can be implemented by creating different types of measures. The present work specifies the coupling kernel to implement the nSML framework due to the following four considerations:

- Kernels are one of the most popular functions to measure the similarity of two objects.
- A kernel can induce a dissimilarity metric straightforwardly.
- Different kernels have the same semantic meaning in the reduced kernel Hilbert space (RKHS), and the combination of them under certain constraints maintains the characteristics of a kernel. This property can be used to leverage different aspects of information by various kernels for heterogeneous data.
- A kernel can be easily adopted into a variety of kernel-based methods for diverse learning tasks: kernel principal component analysis (Mika et al. 1999), kernel clustering (Zhao et al. 2009), relational learning (Frasconi et al. 2014), a Gaussian

process (Rasmussen & Williams 2006), support vector machine (Vapnik 1998), and extreme learning machine (Huang et al. 2012).

In this chapter, a popular type of kernel, namely the squared exponential kernel, has been adopted to observe whether a very common way of instantiating nSML, specifically cKML, can offer significantly improved performance. In the following, cKML is detailed in terms of building an intra-attribute coupling metric, an inter-attribute similarity metric, an object-coupling metric, and a non-IID distance metric. The metric properties are discussed in Section 4.3.

4.2.2 Intra-Attribute Coupling Metric

Intra-attribute couplings represent the interactions among the different values of an attribute. One basic way to observe the intra-attribute couplings of an attribute is to analyze its value distribution. As stated by Boriah et al. (2008) and by Wang, Chi, Zhou & Wong (2015), the value frequency is suitable for describing value distributions. Therefore, a reasonable way to calculate intra-attribute couplings is to measure the similarity between attribute values in terms of value frequency. For example, for a watermelon color shown in Table 2.1, if the number of watermelon with color *yellow* is more similar to the number of watermelon with color *green* rather than that of watermelon with color *white*, then it is argued that *yellow* is closer to *green* rather than to *white*. The results are also consistent with the real watermelon property that the watermelon with color *yellow* is generally close to watermelon with color *green* but differs from that with color *white*.

Hence, a kernelized intra-attribute coupling measure (k_{Ia}) is proposed to capture the intra-attribute couplings in terms of value occurrence frequencies. Measure k_{Ia} involves the attribute value frequency and assumes that two values are more similar if their frequencies are closer. Basically, k_{Ia} can be instantiated according to different frequency distributions. For example, a squared exponential kernel can be used to capture local variation distributions, and a periodic kernel may be adopted to reveal a repeating distribution (Duvenaud et al. 2013). This method uses a squared exponential (Gaussian)-like kernel since it has many good properties and appears as a default kernel in most kernel-based learning methods.

Subsequently, for a categorical data set, given two objects o_i and o_j with attribute values $v_i^{(l)}$ and $v_j^{(l)}$ in the attribute a_l, the intra-attribute coupling measure $k_{Ia}^{(l)}(\mathsf{o}_i, \mathsf{o}_j)$ in

terms of a squared exponential kernel can be formalized as follows:

$$k_{Ia}^{(l)}(\mathsf{o}_i,\mathsf{o}_j) = e^{-(f_j(v_i^{(l)})-f_j(v_j^{(l)}))^2}, \tag{4.1}$$

where $f_j(\cdot)$ is the VFF function that represents the frequency of a specific value of the attribute a_j, as defined in Definition 3.3, and j refers to the j-th attribute.

Taking the watermelon information in Table 2.1 as an example, the similarity between A1 and A2 with respect to attribute *color* is $k_{Ia}^{(2)}(A1,A2) = e^{-(\frac{1}{6}-\frac{2}{6})^2} = e^{-\frac{1}{36}}$, and the one between A2 and A4 is

$$k_{Ia}^{(2)}(A2,A4) = e^{-(\frac{2}{6}-\frac{2}{6})^2} = 1.$$

Since the intra-attribute coupling measure $k_{Ia}^{(j)}(\mathsf{o}_i,\mathsf{o}_j)$ is a positive semi-definite stationary kernel, as will be proved in Section 4.3, ranging from e^{-1} to 1, the intra-attribute coupling kernel can be easily combined with other coupling kernels. Meanwhile, it can effectively induce the corresponding distance metric through a transformation function that will be introduced in Section 4.2.5.

Equation (4.1) assumes that the similarity of attribute values exhibits an exponential decrease when their occurrence frequencies increase. Only if the occurrence frequencies of two attribute values are very similar does their similarity result in a large value; otherwise, their similarity is small and tends to be e^{-1}.

Compared with simple matching-based methods, k_{Ia} not only distinguishes different values but also reserves their local geometric structures. Moreover, differing from existing intra-attribute information-based methods (Boriah et al. 2008, Wang, Chi, Zhou & Wong 2015), k_{Ia} enforces the similarity of values to a quite small and similar value when their frequency difference is larger than a certain threshold. It embeds the value frequency similarity into a value similarity space and eliminates the disturbance from noisy data. For this reason, the proposed intra-attribute coupling measure can achieve reasonable performance.

4.2.3 Inter-Attribute Coupling Metric

The intra-attribute coupling metric represents only the relationships among the different values of an attribute. It does not capture the interactions between attributes. Accordingly, a further metric is proposed, the inter-attribute coupling metric, to capture such interactions. Taking the watermelon information in Table 2.1 as an example, the number of watermelon with color *white* may be similar to that of watermelon with color *black*. If the intra-attribute coupling metric is used alone to reveal the relationship between color *white* and *black*, the similarity between *white* and *black* may result in a high

value. However, this conclusion contradicts the fact that watermelon with color *white* and *black* have very different properties. To address the above problem, inter-attribute couplings are introduced to combine the couplings within an attribute with respect to the semantic information (or knowledge) from and interactions with other attributes.

The inter-attribute coupling metric (k_{Ie}) captures the couplings between values in an attribute with respect to other attributes. Intuitively, the attribute values of an attribute may be seen as similar if they are associated with values in other attributes with similar frequencies. Hence, k_{Ie} takes the difference between the conditional probabilities of two values in an attribute with respect to the values of other attributes to represent the inter-coupling relationship. If two values of an attribute have similar conditional probabilities with respect to the values of other attributes, they are treated as similar.

Let us revisit the case of the watermelon information previously mentioned. If the number of watermenlons with *white* color and watermelons with *black* color are similar, but the values of the other attributes (e.g., the root shape) differ significantly, white watermenlons and black watermelons are still made distinguishable by involving the additional attribute $rootshape$ when considering conditional probability.

The cKML method implements the inter-attribute coupling metric k_{Ie} by using a squared exponential kernel. It first captures the couplings between each attribute pair, and it then extends the pairwise attribute couplings to all attributes. Given a categorical data set, the coupling kernel $k_{Ie}^{(l,m)}(\mathsf{o}_i,\mathsf{o}_j)$ for the l-th attribute with respect to the m-th attribute is defined as follows:

$$k_{Ie}^{(l,m)}(\mathsf{o}_i,\mathsf{o}_j) = e^{-z^2},$$

and

$$z = \sum_{\mathsf{o}_q \in \cup_m} p_{v_q^{(m)}} \cdot |p_{l|m}(v_q^{(m)}|v_i^{(l)}) - p_{l|m}(v_q^{(m)}|v_j^{(l)})|,$$

where $p_{l|m}(\cdot|\cdot)$ is the ICPF function as defined in Definition 3.4, $\mathsf{o}_q \in \cup_m$ collects the objects whose value in m-th attribute has co-occurrence with either $v_i^{(l)}$ or $v_j^{(l)}$ in the attribute a_l, and $P_{v_q^{(m)}}$ is the occurrence probability of value $v_q^{(m)}$ in set $\cup_m$.

Taking the watermelon information in Table 2.1 as an example, the similarity between the attribute *color* of *A1* and *A6* with respect to the attribute *root shape* is $k_{Ie}^{(2,3)}(A1,A6) = e^{-(\frac{1}{2}|\frac{1}{1}-\frac{0}{1}|+\frac{1}{2}|\frac{0}{1}-\frac{1}{1}|)^2} = e^{-1}$, and the similarity between the attribute *root shape* of *A2* and *A5* with respect to the attribute *color* is

$$k_{Ie}^{(3,2)}(A2,A5) = e^{-(\frac{1}{4}|\frac{1}{2}-\frac{0}{2}|+\frac{2}{4}|\frac{1}{2}-\frac{1}{2}|+\frac{1}{4}|\frac{0}{2}-\frac{1}{2}|)^2} = e^{-\frac{1}{16}}.$$

The accumulative inter-attribute coupling kernel $k_{Ie}^{(l)}(\mathsf{o}_i, \mathsf{o}_j)$ for the l-th attribute is defined as in Equation (4.2). It aggregates the pairwise attribute interactions between the j-th attribute and all others.

$$k_{Ie}^{(l)}(\mathsf{o}_i, \mathsf{o}_j) = \sum_{m=1, m\neq l}^{n_a} \alpha_m k_{Ie}^{(l,m)}(\mathsf{o}_i, \mathsf{o}_j), \tag{4.2}$$

where n_a is the number of attributes. Equation (4.2) uses the summation of the couplings between each attribute pair to construct the inter-attribute couplings. This is because the target of k_{Ie} is to reveal the possible connections between two attribute values in terms of different dimensions. If an attribute pair has strong couplings in any dimension, it can be regarded as having strong inter-attribute coupling. In practice, it is worth noting that not all attribute pairs have a strong relationship, and the degree of attribute-pair couplings may also be different. Therefore, a summation coefficient α_m is added to fit such situations. The summation coefficient α_m is determined in a similar way as in Wang, Chi, Zhou & Wong (2015) and Ienco et al. (2012).

For the attribute pair $(\mathsf{a}_l, \mathsf{a}_m)$, the symmetric uncertainty (SU) is used to measure their relevance, which is defined as:

$$SU_{\mathsf{a}_l}(\mathsf{a}_m) = 2 \times \frac{H(\mathsf{a}_l) - H(\mathsf{a}_l|\mathsf{a}_m)}{H(\mathsf{a}_l) + H(\mathsf{a}_m)},$$

where $H(\mathsf{a}_l)$ and $H(\mathsf{a}_l|\mathsf{a}_m)$ are the entropy and condition entropy of variables a_l and a_m respectively. α_m is set as $SU_{\mathsf{a}_l}(\mathsf{a}_m)$ because $SU_{\mathsf{a}_l}(\mathsf{a}_m)$ reveals the degree of inter-attribute couplings. To avoid those cases where an attribute pair does not hold couplings, here added one more constraint that α_m is set to 0 if the following two conditions are both satisfied: (1) $SU_{\mathsf{a}_l}(\mathsf{a}_m) \leq SU_{\mathsf{a}_l}(\mathsf{a}_q)$; and (2) $SU_{\mathsf{a}_q}(\mathsf{a}_m) \geq SU_{\mathsf{a}_l}(\mathsf{a}_m)$.

Let $Sum(\alpha) = \sum_{m,m\neq l}^{n_a} \alpha_m$. If $Sum(\alpha) \neq 0$, then α_m is normalized by $\alpha_m / Sum(\alpha)$. Otherwise, $k_{Ie}^{(l)}(\mathsf{o}_i, \mathsf{o}_j)$ is set to be e^{-1} since there is no significant coupling between attributes.

The inter-attribute metric k_{Ie} is also a positive semi-definite stationary kernel, ranging from e^{-1} to 1, which will be proved in Section 4.3. According to the transformation function proposed in Section 4.2.5, an inter-coupling distance metric can be induced from k_{Ie}.

The inter-attribute coupling metric implemented in this chapter has three notable advantages, which make the proposed inter-attribute coupling metric significantly better than existing inter-attribute-based similarity measures as discussed in the following.

The first merit is that it can reveal more comprehensive contextual information compared to the previous work. k_{Ie} adopts a union value set instead of an intersection value

set as the context, but the existing work uses either an intersection value set (Ienco et al. 2012, Wang, Chi, Zhou & Wong 2015) or all the values in the other attributes (Jia et al. 2016) as the context. Given two values from an attribute, the union value set refers to the values in the other attributes that co-occur with any of the given values. Meanwhile, the intersection value set refers to the values in other attributes that co-occur with both of the given values. If the context is set as the intersection value set, the contribution from values in another attribute, which only co-occur with one compared value, will be ignored. For instance, in the previous scenario, the similarity between *Mary* and *Alice* in terms of *Commitment* conditional on *Performance* will be 1 because $high$ and $intermediate$ have the same occurrence frequency with the intersection value $rank\ B$. However, $high$ also co-occurs with $rank\ A$ while $intermediate$ co-occurs with $rank\ C$, which indicates their difference. If all values in another attribute are used as the context, the similarity will be affected by unrelated values. Particularly, a large number of unrelated values may even cause wrong results. Fortunately, using the union value set as the context can overcome such drawbacks and comprehensively reflect inter-attribute couplings.

The second benefit is that k_{Ie} can eliminate both the low-level noise in attribute pairs and the high-level noise caused by combining uncoupled attributes. Although some of the existing work (Ienco et al. 2012, Wang, Chi, Zhou & Wong 2015) can reduce high-level noise by introducing the SU, they cannot filter low-level noise within attribute pairs which is commonly caused by sampling and randomness. k_{Ie}, on the other hand, can remove this noise through projecting the similarity with small co-occurrence frequency into a quite similar and small value.

The third advantage is that k_{Ie} embeds the co-occurrence similarity into the object similarity using the nearest neighbor assumption which has a solid foundation according to manifold learning (Belkin et al. 2006). Even though the previous work introduced co-occurrence similarity to represent object similarity, they did not provide a sound explanation regarding why co-occurrence similarity can be transformed to object similarity.

4.2.4 Object Coupling Metric

So far, it is shown that the proposed attribute couplings can capture two types of attribute-oriented interactions, i.e., intra-attribute couplings and inter-attribute couplings. This section proposes the concept of *object coupling* to facilitate an understanding of object interactions based on the above comprehensive low-level attribute couplings.

The object coupling determines whether some objects are more similar than others. It aggregates low-level couplings from different attributes. Here measures it by proposing the object coupling kernel k_O. The object coupling kernel integrates both the different low-level couplings of an attribute and the attribute couplings from different attributes.

With the integration of different low-level couplings, k_O adopts the multiplication for the former to combine two aspects of attribute couplings, i.e., intra-attribute couplings and inter-attribute couplings. It reflects the following observations: (1) the couplings between two objects are contributed by different factors (e.g., value and attribute) and from different interaction aspects (e.g., value-value interactions and attribute-attribute interactions), and (2) the level of overall object couplings is determined by the most distinguishable aspect. Regarding the second observation, the object couplings will be smaller than the weakest low-level coupling. Since the smaller coupling reflects the stronger distinguishable information, the object couplings will reserve the most useful information for learning.

For the integration of attribute couplings from different attributes, couplings from different attributes are added together to obtain object couplings. This follows the assumption that the couplings between two objects are contributed by all attributes. The aggregated object coupling kernel $k_O(\mathsf{o}_i, \mathsf{o}_j)$ between two objects o_i and o_j is calculated on all n_a attributes in a data set as follows,

$$k_O(\mathsf{o}_i, \mathsf{o}_j) = \sum_{l=1}^{n_a} \beta_l \cdot k_{Ia}^{(l)}(\mathsf{o}_i, \mathsf{o}_j) \cdot k_{Ie}^{(l)}(\mathsf{o}_i, \mathsf{o}_j), \tag{4.3}$$

where β_l is the combination coefficient that reflects the contribution from l-th attribute.

This thesis makes the following three assumptions regarding β_l: (1) each attribute makes a different contribution to the final object similarity; (2) each coupling aspect makes a different contribution; and (3) the more distinguishable information an attribute contains, the larger the contribution it makes. Under these assumptions, we give a weighting criterion: the attribute weight is assigned according to the probability that two compared objects own different information. The intuition of this criterion is that more attention should be paid to the differential information which is the foundation of discrimination capability. With this criterion, a data-driven method for dynamic weighting is proposed to jointly consider intra- and inter-attribute couplings.

Regarding the intra-attribute couplings, the differential information is considered as the probability of two objects owning different attribute values. This probability reflects to what extent an attribute can distinguish two objects. A small probability illustrates

poor differentiation capability, and vice versa. Inspired by Jia et al. (2016), for the l-th attribute, this probability $p_{Ia}^{(l)}$ can be calculated as follows:

$$p_{Ia}^{(l)} = 1 - \sum_{q=1}^{n_v^{(l)}} f_l(\mathsf{v}_q^{(l)}) f_l^-(\mathsf{v}_q^{(l)}),$$

where $f_l(\cdot)$ is the VFF defined in Definition 3.3, and $f_l^-(\cdot)$ has a similar definition as $f_l(\cdot)$ yet it removes an object with value $\mathsf{v}_q^{(l)}$ in the population as in Equation (4.4).

$$f_l^-(\mathsf{v}_q^{(l)}) = \frac{|g_l(\mathsf{v}_q^{(l)})| - 1}{n_o - 1}, \tag{4.4}$$

where n_o is the number of all objects in the data set. Subsequently, the normalized contribution from intra-attribute couplings for each attribute can be assigned as:

$$\beta_{Ia}^{(l)} = \frac{p_{Ia}^{(l)}}{\sum_{m=1}^{n_a} p_{Ia}^{m}}.$$

Regarding the inter-attribute couplings, the differential information can be revealed by co-occurring value pairs. For two attributes, the distinguishing capability is large when the probability of any co-occurrence value pair is small. To this end, here calculates the probability of the co-occurring value pair to capture this differential information. By labelling the co-occurring value pair set as

$$CO_{lm} = \{joint_{lm,1}, joint_{lm,2}, \cdots, joint_{lm,n_{lm}}\},$$

the probability $p_{Ie}^{(l,m)}$ is formulated as:

$$p_{Ie}^{(l,m)} = 1 - \sum_{q=1}^{n_{lm}} f_{lm}(joint_{lm,q}) f_{lm}^-(joint_{lm,q}).$$

Due to the fact that an attribute can associate with any other attributes, the probability of these associations should be aggregated to construct the weight for inter-attribute couplings. Here uses summation with the coefficient α in Equation (4.2) to aggregate these probabilities as follows,

$$p_{Ie}^{(l)} = \sum_{m=1, m\neq l}^{n_a} \alpha_m p_{Ie}^{(l,m)}.$$

After this, the normalized contribution from inter-attribute couplings for each attribute can be calculated as,

$$\beta_{Ie}^{(l)} = \frac{p_{Ie}^{(l)}}{\sum_{m=1}^{n_a} p_{Ie}^{(m)}}.$$

The contribution of each attribute is defined by jointly considering the contribution of intra- and inter-attribute couplings,

$$\beta_u^{(l)} = \max\{\beta_{Ia}^{(l)}, \beta_{Ie}^{(l)}\}.$$

Then, the normalized combination coefficient $\beta^{(l)}$ can be calculated as follows,

$$\beta^{(l)} = \frac{\beta_u^{(l)}}{\sum_{m=1}^{n_a} \beta_u^{(m)}}.$$

Here uses the maximum value of $\beta_{Ia}^{(l)}$ and $\beta_{Ie}^{(l)}$ as the attribute contribution due to the fact that the overall object couplings are determined by the weakest low-level couplings which have the largest distinguishable contribution.

Note that $k_O(\mathsf{o}_i, \mathsf{o}_j)$ is also positive semi-definite, since it is composed of the multiplication of three positive definite kernels (as shown in Lemma 4.3). Since $k_O(\mathsf{o}_i, \mathsf{o}_j)$ captures hierarchical couplings between objects, it has much richer and more comprehensive interactions between objects than the existing similarity representations. In fact, the object coupling metric transfers the original data represented by its original feature space to a hierarchical coupling-based kernel space. The transformed kernel-based spaces (1) are embedded with multi-level and multi-aspect couplings that reflect the substantial low-level interactions in data, (2) enable the integration of these couplings that contribute to the representation of data characteristics and complexities, and (3) combine low-level data characteristics with high-level model-based power (in this chapter, squared exponential kernel as the model) to describe data characteristics. In the transformed kernel-based high-dimensional spaces, objects are relocated and represented in terms of hierarchical coupling relationships. As a given data set is often embedded with different forms and characteristics of coupling relationships (Cao 2015), the similarity learning objective is to capture them as much and as deeply as possible. For this, the corresponding kernel functions can be identified to catch and integrate the sophisticated couplings to model object coupling-based similarity. In Section 4.4.4, the further research is to discuss the challenges and prospects of selecting appropriate kernels and models to represent such couplings in complex data.

With the kernel properties analyzed in Section 4.3, the proposed coupled kernel metric can be applied to diverse kernel-based learning machines (e.g., support vector machine, a Gaussian process, kernel k-means, and extreme learning machine) for a variety of learning tasks. In doing this, it is important to select the most appropriate kernels such that the intrinsic data complexities can be well captured for specific learning objectives, as discussed in Section 4.4.4.

4.2.5 Non-IID Distance Metric

This section induces a distance metric from the proposed non-IID similarity metrics. Given two objects o_i and o_j and a proposed non-IID similarity metric $k(\cdot,\cdot)$, such as the intra-attribute coupling metric (Equation (4.1)), inter-attribute coupling metric (Equation (4.2)), or the object coupling metric (Equation (4.3)), a non-IID distance metric $d(\cdot,\cdot)$ can be constructed by Equation (4.5).

$$d(o_i, o_j) = \sqrt{k(o_i, o_i) - 2k(o_i, o_j) + k(o_j, o_j)}. \tag{4.5}$$

As seen in Section 4.3, this distance metric maintains all metric properties, and its value ranges from $[0, \sqrt{2-2e^{-1}}]$.

4.3 Theoretical Analysis of Non-IID Similarity Metrics Learning Properties

This section first proves the positive semi-definite properties of the proposed coupled kernel and the metric properties of the induced non-IID distance. It then analyzes the time complexity of the proposed coupled kernel.

4.3.1 The Positive Semi-Definite Property of Coupled Kernels

It is claimed that the proposed coupling metrics in the previous sections satisfy the positive semi-definite properties. Retaining the positive semi-definite property is critical for the soundness and other merits of the proposed coupling measures. If and only if this property is satisfied, the coupling metrics can be used in kernel-based analysis, such as support vector machine and kernel k-means. In addition, this property ensures that different kinds of similarities can be integrated through the closure operations in RKHS, which forms the foundation of the proposed nSML framework in Figure 4.1.

Before proven that the object coupling metric is positive semi-definite, three foundational lemmas are introduced and prove the positive semi-definite property of the intra-attribute coupling metric and the inter-attribute coupling metric.

Lemma 4.1. *If $f(\cdot)$ is a real value function defined on $\mathcal{X} \subset \mathbb{R}^n$, then $k(x,y) = f(x)f(y)$ is a positive semi-definite kernel.*

Lemma 4.2. *If $k_1(x,y)$ is a positive semi-definite kernel in $\mathcal{X} \times \mathcal{X}$, then $k(x,y) = e^{k_1(x,y)}$ is a positive semi-definite kernel.*

Lemma 4.3. *If $k_1(x,y)$ and $k_2(x,y)$ are positive semi-definite kernels in $\mathcal{X} \times \mathcal{X}$, and a constant $a \geq 0$, then the following functions are positive semi-definite kernels:*

1. $k(x,y) = k_1(x,y) + k_2(x,y)$;
2. $k(x,y) = ak_1(x,y)$;
3. $k(x,y) = k_1(x,y)k_2(x,y)$.

The proofs of these three lemmas can be found in Section 4.1 in Steinwart & Christmann (2008). Based on the above three lemmas, an additional lemma is proposed as follows.

Lemma 4.4. $k_1(\mathsf{o}_i,\mathsf{o}_j) = e^{-f_l(v_i^{(l)})}e^{-f_l(v_j^{(l)})}$ *and* $k_2(\mathsf{o}_i,\mathsf{o}_j) = 2f_l(v_i^{(l)})f_l(v_j^{(l)})$ *are positive semi-definite kernels.*

Proof. Let $f(v_j^{(l)}) = e^{-f_l(v_j^{(l)})}$ or $f(v_j^{(l)}) = \sqrt{2}f_l(v_j^{(l)})$, Lemma 4.4 can be straightforwardly induced from Lemma 4.1. ■

Lemma 4.5. *Given a kernel* $k_3(\mathsf{o}_i,\mathsf{o}_j) = e^{2f_l(v_i^{(l)})f_l(v_j^{(l)})}$, *the kernel is positive semi-definite.*

Proof. Since $k_2(\mathsf{o}_i,\mathsf{o}_j) = 2f_l(v_i^{(l)})f_l(v_j^{(l)})$ is a positive semi-definite kernel, according to Lemma 4.2, $k_3(\mathsf{o}_i,\mathsf{o}_j) = e^{k_2(\mathsf{o}_i,\mathsf{o}_j)}$ is positive semi-definite. ■

Theorem 4.1. *The intra-attribute coupling kernel $k_{Ia}^{(l)}(\mathsf{o}_i,\mathsf{o}_j)$ proposed in Equation (4.1) is a positive semi-definite kernel.*

Proof. Since

$$k_{Ia}^{(l)}(\mathsf{o}_i,\mathsf{o}_j) = e^{-(f_l(v_i^{(l)})-f_l(v_j^{(l)}))^2} = e^{-f_l(v_i^{(l)})}e^{-f_l(v_j^{(l)})}e^{2f_l(v_i^{(l)})f_l(v_j^{(l)})} = k_1(\mathsf{o}_i,\mathsf{o}_j)k_3(\mathsf{o}_i,\mathsf{o}_j),$$

and kernels $k_1(\mathsf{o}_i,\mathsf{o}_j)$ and $k_3(\mathsf{o}_i,\mathsf{o}_j)$ are positive semi-definite, according to Lemma 4.3, $k_{Ia}^{(l)}(\mathsf{o}_i,\mathsf{o}_j)$ is a positive semi-definite kernel. ■

Theorem 4.2. *The inter-attribute coupling kernel $k_{Ie}^{(l)}(\mathsf{o}_i,\mathsf{o}_j)$ proposed in Equation (4.2) is a positive semi-definite kernel.*

Proof. Since $k_{Ie}^{(l,m)}$ takes the same form as the intra-attribute coupling kernel, which is positive semi-definite, $k_{Ie}^{(l,m)}$ is positive semi-definite. According to Lemma 4.3, the weighted summation of valid kernels is still a valid kernel. Hence, $k_{Ie}^{(l)}(\mathsf{o}_i,\mathsf{o}_j)$ is a positive semi-definite kernel. ■

Lastly, Theorem 4.3 is proposed and proved.

Theorem 4.3. *The object coupling kernel* $k_O(\mathsf{o}_i, \mathsf{o}_j) = k_{Ia}(\mathsf{o}_i, \mathsf{o}_j) \cdot k_{Ie}(\mathsf{o}_i, \mathsf{o}_j)$ *is a positive semi-definite kernel.*

Proof. Since k_{Ia} and k_{Ie} are both positive semi-definite, the object coupling kernel k_O is consequently positive semi-definite, according to Lemma 4.3. ■

4.3.2 Stationary and Metric Properties of Coupled Kernels

In this section, this thesis further discuss the *stationary property* and the *metric properties* held by the object coupling kernel.

Proposition 4.1. ***Stationary Property.*** *The proposed object coupling kernel is a stationary kernel; that is,* $k_O(x, y) = \hat{k}(|x - y|)$.

Proof. The thesis first proves that the compositions of the object coupling kernels hold the stationary property. With respect to the intra- and inter-attribute coupling kernels, their stationary property is guaranteed by the square operation. Subsequently, the coupled kernel is stationary because it is composed of stationary kernels through multiplication. ■

With the stationary property, a kernel assigns the same similarity to value pairs, which have the same value difference. In other words, it is invariant of the translation of input values when their value gap is kept the same. This property guarantees that the non-IID similarity metric captures the low-level coupling relationships without bias. Without this property, a similarity may implicitly involve an assumption of specific data distribution. For example, the linear kernel, which does not obey this property, gives a larger value to the pair (5, 7) than to (2, 4), which implies a denser data distribution for a larger value.

Proposition 4.2. ***Metric Properties.*** *The proposed non-IID similarity metrics can induce a non-IID distance measure* $d(\mathsf{o}_i, \mathsf{o}_j)$ *which satisfies three conditions of a metric:*

1. *triangle inequality:* $d(\mathsf{o}_i, \mathsf{o}_k) + d(\mathsf{o}_k, \mathsf{o}_j) \geq d(\mathsf{o}_i, \mathsf{o}_j)$*;*
2. *commutativity:* $d(\mathsf{o}_i, \mathsf{o}_j) = d(\mathsf{o}_j, \mathsf{o}_i)$*; and*
3. *reflexivity: if* $\mathsf{o}_i = \mathsf{o}_j$*, then* $d(\mathsf{o}_i, \mathsf{o}_j) = 0$.

Proof. The distance $d(\mathsf{o}_i, \mathsf{o}_j)$ can be constructed as in Equation (4.5). Since the object coupling measure $k_O(\mathsf{o}_i, \mathsf{o}_j)$ is a positive semi-definite kernel, its corresponding distance satisfies the above metric properties according to *Proposition 5.2* in Gärtner et al. (2004). ■

The above metric properties are important features of non-IID similarity metrics. With these properties, the proposed kernel-based coupling measures can be transferred to a distance metric, reflecting the sample distance in RKHS. Furthermore, these properties guarantee the nSML framework can induce a valid distance, which can be applied to distance-based methods such as k-nearest neighbors (KNN) and k-modes.

4.3.3 Time Complexity of Coupled Kernel Metrics Learning

Two data factors affect the time complexity of coupling measures. They are the number of attributes (n_a) and the maximum number of values for each attribute (n_{mv}).

For the intra-attribute coupling kernel, the value pair similarity should be calculated for all attributes. Therefore, the time complexity of calculating $k_{Ia}(\cdot,\cdot)$ is $O(n_a n_{mv}^2)$. For the inter-attribute coupling kernel, the largest time cost is calculating the value pair similarity in terms of their union values in another attribute for all attribute pairs. Thus, the time complexity of calculating $k_{Ie}(\cdot,\cdot)$ is $O(n_a^2 n_{mv}^3)$. Because the object coupling kernel combines these two low-level coupling measures, the time complexity of k_O is determined by the largest time cost $O(n_a^2 n_{mv}^3)$.

As shown above, the maximum number of values for each attribute has a greater impact on time complexity than does the number of attributes. Moreover, the intra-attribute coupling kernel has smaller time complexity compared to the inter-attribute coupling kernel and the object coupling kernel. If n_a or n_{mv} is large, calculating k_{Ia} will be significantly faster than calculating k_{Ie} and k_O. Hence, when a trade-off between accuracy and time cost is needed, k_{Ia} can substitute k_O for large data sets.

4.4 Experiments and Evaluation of Non-IID Similarity Metrics Learning Performance

This section evaluates the learning performance, learning quality and the computational performance of cKML by empirical analysis. Regarding the learning performance, cKML is feed into two popular learners to illustrate its effect: k-modes for clustering

and KNN for classification. For learning quality, the cKML-represented categorical data is quantitatively evaluated by information retrieval and qualitatively visualized by the data distribution in the cKML-learned metric space. For computational performance, the synthetic data sets with different data factors are adopted to test the scalability of cKML.

4.4.1 Testing Non-IID Similarity Metrics-Enabled Learning Performance

In order to analyze a large data set, two typical learning methods with good scalability are chosen. For clustering, k-modes is chosen, while KNN is selected for supervised classification. It is worth noting that cKML can also be used in other scenarios, such as for recommender systems, outlier detection, and regression.

The evaluation of cKML performance is undertaken over the following steps: (1) The proposed coupled metric is compared with two baseline categorical distance measures, Hamming Distance ("Hamming", for short) and OF (Boriah et al. 2008); (2) it is also compared with four state-of-the-art similarity and dissimilarity measures for categorical data clustering: Ahmad (Ahmad & Dey 2007), Rough (Cao, Liang, Li, Bai & Dang 2012), DILCA (Ienco et al. 2012), and COS (Wang, Dong, Zhou, Cao & Chi 2015), to determine whether cKML is superior in representing similarity; (3) the low-level couplings (i.e., the intra- and inter-attribute couplings) are compared with each other and compared with high-level couplings to determine whether hierarchical coupled similarity can improve learning accuracy.

4.4.1.1 Coupled Kernel Metric Learning-Enabled *K*-Modes Clustering

The method k-modes is an extended version of k-means for categorical data clustering. It uses the mode instead of the mean value of attribute values to calculate the clustering center. It originally utilizes Hamming distance to measure the distance between objects and clustering centers. In this experiment, Hamming distance is replaced with other categorical distance measures to test their performance.

Since the k-modes algorithm is sensitive to the initial center selection, it is run 100 times with the randomly selected initial centers. The clusters with the minimum objective function value obtained over 100 runs are selected to demonstrate the performance.

The clustering performance is shown in Table 4.1 with respect to F-score and in Table 4.2 with respect to NMI. In these tables, the best result for each data set is shown in

bold face, and the overall performance with respect to all data sets is given in the bottom row using the mean value and AR. The results show that the proposed cKML-enabled k-modes has the best overall F-score performance (0.6126), which is an improvement of **0.0161** over the best state-of-the-art method, DILCA, and an improvement of **0.0510** over the best baseline method, Hamming, in terms of F-score (see Table 4.1). The cKML method also obtains the best overall NMI performance (0.3586), i.e., improving **0.0186** over the best state-of-the-art method, DILCA, and **0.0845** over the best baseline method, Hamming, in terms of NMI. Meanwhile, the cKML method achieves the best clustering performance in 11 and 13 data sets in terms of F-score and NMI, respectively. On the contrary, the other methods mostly achieve the best performance in 5 and 7 data sets under these two criteria. As a result, cKML achieves an AR of **2.6250** in terms of F-score and **2.2222** in terms of NMI, which are significantly better than the competitors. Furthermore, it is worth noting that the largest performance improvement of cKML from the other methods is **0.2521** with respect to F-score and is **0.3191** with respect to NMI in the *Spl* data set.

In order to examine the contribution and learning effect made by measuring low-level couplings, the clustering performance is compared when involving each low-level coupling metric learning (i.e., intra-attribute coupling kernel learning [intra-cKML] and inter-attribute coupling kernel learning [inter-cKML]) and the object coupling metric learning (i.e., cKML). Furthermore, these measures are compared with their counterparts (i.e., intra-COS, inter-COS, and COS) in the COS similarity system (Wang, Dong, Zhou, Cao & Chi 2015), since COS is a similar state-of-the-art hierarchical similarity measure, to demonstrate the superiority of cKML.

The results are shown in Tables 4.3 and 4.4 in terms of F-score and NMI, respectively. In these tables, different measures are compared separately in terms of the couplings captured by each of them (i.e., the intra-attribute coupling, the inter-attribute coupling, and the object coupling). The bold typeface indicates the best result in a given measure, and the overall performance is also given as a mean value and an AR. As the overall performance illustrates, the intra-cKML significantly improves the performance from 0.2822 to 0.6019 (an improvement of **0.3197**) compared with intra-COS in terms of F-score; similarly, the NMI performance is improved from 0.0238 to 0.3359 (gaining **0.3121**). However, inter-cKML only slightly improves the performance from 0.5674 to 0.5717 (improving **0.0043** only) in terms of F-score and from 0.3071 to 0.3168 (gaining **0.0097**) in terms of NMI. Regarding the object coupling measure, cKML achieves 0.6126 and 0.3586 with respect to F-score and NMI, respectively. However, the COS measure

Table 4.1: K-modes Clustering F-score with Different Similarity Measures

Data Set	**cKML**	**COS**	**Ahmad**	**DILCA**	**Rough**	**Hamming**	**OF**
SoyS	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9805	0.9805	0.9250
Zoo	0.7410	0.7210	0.7134	0.7134	0.6279	0.7327	**0.7805**
DNAP	**0.8962**	0.4924	0.4992	0.8585	0.6320	0.5268	0.5178
Hay	**0.4060**	0.3898	0.3376	0.3287	0.3892	0.3306	0.3911
Lym	0.3868	**0.5581**	0.4036	0.3634	0.3231	0.3265	0.2444
Hep	0.6448	0.4629	0.6672	0.6513	0.5921	0.5921	**0.6769**
Aud	0.3381	0.2771	**0.3538**	0.3177	0.2236	0.2905	0.3207
Hsv	0.8836	0.8836	0.8836	**0.8879**	0.8704	0.8664	0.8664
Spc	0.3510	0.3626	0.3493	0.3476	**0.5763**	0.3594	0.3594
Mof	0.4843	0.5018	0.5022	0.4868	0.5062	**0.5098**	**0.5098**
SoyL	0.5846	**0.6010**	0.5684	0.5942	0.4641	0.5531	0.5032
Prim	0.2495	0.1981	0.2365	0.2176	0.2238	**0.2619**	0.1944
Dmg	0.7448	**0.7458**	0.7287	0.7261	0.5799	0.6660	0.3321
Wcs	**0.9666**	0.9428	0.9512	0.9549	0.9444	0.8998	0.4876
Crx	0.5265	0.3699	0.5265	**0.7929**	0.6347	**0.7929**	0.7365
Br	**0.9641**	0.9356	0.9489	0.9525	0.9437	0.9327	0.4860
Ma	0.8204	0.8006	0.8166	**0.8265**	0.8067	0.8150	0.8067
Tic	**0.5903**	0.5188	0.5087	0.5297	0.5019	0.5359	0.4883
Flr	0.3476	0.3579	0.3420	0.3559	0.3885	0.3922	**0.5376**
Titn	0.3372	0.2977	0.3372	0.3372	**0.3627**	0.3372	0.3372
DNAN	**0.7097**	0.4191	0.4668	0.5918	0.4328	0.4144	0.3996
Spl	**0.7108**	0.3131	0.4734	0.4587	0.4279	0.4248	0.4054
Krv	0.5520	0.4672	0.5517	0.5517	0.5373	0.5386	**0.5682**
Ld	**0.6494**	0.5391	0.5183	0.6108	0.3265	0.2882	0.1970
Ms	0.8281	**0.8291**	0.8286	0.8239	0.7818	0.8229	0.8133
Adt	**0.6830**	0.6007	0.6776	0.6109	0.6016	0.6112	0.5957
Cnt	**0.3370**	0.2723	0.3288	0.3314	0.3034	0.3143	0.2988
Mean	**0.6102**	0.5433	0.5669	0.5920	0.5584	0.5592	0.5100
AR	**2.6250**	4.6071	3.6607	3.4821	4.5893	4.2143	4.8214

Table 4.2: K-modes Clustering NMI with Different Similarity Measures

Data Set	cKML	COS	Ahmad	DILCA	Rough	Hamming	OF
SoyS	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9407	0.9407	0.8565
Zoo	0.8403	**0.8708**	0.8043	0.8043	0.7423	0.7538	0.7870
DNAP	**0.5196**	0.0000	0.0012	0.4121	0.0510	0.0023	0.0010
Hay	**0.0284**	0.0209	0.0005	0.0000	0.0105	0.0021	0.0107
Lym	**0.1844**	0.1536	0.1695	0.1337	0.1246	0.1231	0.0702
Hep	0.1457	0.0000	**0.1633**	0.1158	0.1271	0.1271	0.1219
Aud	0.5283	0.4601	0.5068	**0.5423**	0.4337	0.4769	0.4232
Hsv	0.5111	0.5111	0.5111	**0.5217**	0.4462	0.4462	0.4462
Spc	0.1021	0.0927	0.1035	**0.1049**	0.0811	0.0848	0.0848
Mof	0.0055	0.0099	0.0088	0.0059	0.0037	**0.0104**	**0.0104**
SoyL	0.6924	0.6764	0.6749	**0.7460**	0.6608	0.6600	0.6069
Prim	**0.3690**	0.3344	0.3481	0.3633	0.3238	0.3670	0.3170
Dmg	0.8510	0.8470	**0.8585**	0.8440	0.6187	0.7085	0.2748
Wcs	**0.7944**	0.6881	0.7216	0.7376	0.6951	0.5625	0.0004
Crx	0.0215	0.0013	0.0215	**0.2659**	0.0724	**0.2659**	0.1696
Br	**0.7835**	0.6597	0.7117	0.7286	0.6918	0.6475	0.0006
Ma	0.3286	0.2958	0.3250	**0.3386**	0.3068	0.3291	0.3068
Tic	**0.0265**	0.0029	0.0015	0.0044	0.0005	0.0083	0.0000
Flr	0.3171	0.3575	0.3077	0.4004	0.2646	0.3218	**0.5299**
Titn	0.1038	0.1039	0.1038	0.1038	**0.1249**	0.1038	0.1038
DNAN	**0.3921**	0.0580	0.0745	0.2940	0.0546	0.0542	0.0464
Spl	**0.3926**	0.0133	0.0735	0.0491	0.0535	0.0499	0.0416
Krv	**0.0233**	0.0034	0.0157	0.0157	0.0052	0.0101	0.0223
Ld	**0.5094**	0.4303	0.4344	0.5055	0.1963	0.1346	0.1001
Ms	0.4209	**0.4396**	0.4288	0.3980	0.2593	0.3928	0.3725
Adt	0.1441	0.0778	**0.1457**	0.0822	0.0686	0.0799	0.0944
Cnt	**0.0026**	0.0001	0.0015	0.0013	0.0004	0.0010	0.0016
Mean	**0.3718**	0.3003	0.3155	0.3526	0.2725	0.2839	0.2148
AR	**2.2222**	4.3519	3.3333	3.0370	5.2407	4.5370	5.2778

only achieves 0.5440 for F-score and 0.2897 for NMI. The performance gaps between these measures in terms of F-score and NMI are 0.0686 and 0.0689. The results also demonstrate that, for cKML, the learning performance of the object coupling metric is better than the low-level coupling metrics. However, the COS object similarity measure leads to lower performance. Actually, the overall performance of inter-COS is better than that of COS, as discussed in Section 4.4.4.

Table 4.3: F-score for K-modes Clustering with Low-level cKML and COS Measures

Data Set	**Intra-cKML**	**Intra-COS**	**Inter-cKML**	**Inter-COS**	**cKML**	**COS**
SoyS	**0.9797**	0.1848	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Zoo	**0.7265**	0.1439	**0.7491**	**0.7491**	**0.7410**	0.7210
DNAP	**0.8575**	0.3540	0.5071	**0.5090**	**0.8962**	0.4924
Hay	**0.4051**	0.2094	**0.3628**	0.3661	**0.4060**	0.3898
Lym	0.3601	**0.4459**	0.3750	**0.3929**	0.3868	**0.5581**
Hep	**0.7031**	0.4404	0.6273	**0.6672**	**0.6448**	0.4629
Aud	**0.3070**	0.2429	**0.2622**	0.2145	**0.3381**	0.2771
Hsv	**0.8358**	0.3465	**0.8836**	**0.8836**	**0.8836**	**0.8836**
Spc	0.3817	**0.4414**	**0.3526**	0.3493	0.3510	**0.3626**
Mof	**0.5206**	0.4434	0.5014	**0.5095**	0.4843	**0.5018**
SoyL	**0.5872**	0.1261	0.5625	**0.5861**	0.5846	**0.6010**
Prim	**0.2318**	0.0447	0.2375	**0.2720**	**0.2495**	0.1981
Dmg	**0.6909**	0.0987	0.7288	**0.7343**	0.7448	**0.7458**
Wcs	**0.9374**	0.3985	**0.9679**	0.9512	**0.9666**	0.9428
Crx	**0.5265**	0.3605	**0.5265**	**0.5265**	**0.5265**	0.3699
Br	**0.9325**	0.4003	**0.9572**	0.9489	**0.9641**	0.9356
Ma	**0.7952**	0.3424	0.8153	**0.8166**	**0.8204**	0.8006
Tic	**0.5703**	0.3948	**0.4910**	0.4897	**0.5903**	0.5188
Flr	**0.3565**	0.0841	**0.4699**	0.3420	0.3476	**0.3579**
Titn	**0.2865**	0.1468	**0.3372**	**0.3372**	**0.3372**	0.2977
DNAN	**0.7760**	0.2276	0.4508	**0.4582**	**0.7097**	0.4191
Spl	**0.8362**	0.2286	**0.4870**	0.4654	**0.7108**	0.3131
Krv	**0.5908**	0.3438	**0.5563**	0.5517	**0.5520**	0.4672
Ld	**0.6240**	0.0216	**0.6144**	0.5759	**0.6494**	0.5391
Ms	**0.7936**	0.3825	**0.8291**	0.8286	0.8281	**0.8291**
Adt	**0.5685**	0.4289	**0.6851**	0.6776	**0.6830**	0.6007
Cnt	**0.3203**	0.2648	0.3166	**0.3305**	**0.3370**	0.2723
Mean	**0.6108**	0.2796	**0.5795**	0.5750	**0.6195**	0.5501
AR	**1.0714**	1.9286	**1.4821**	1.5179	**1.2857**	1.7143

Table 4.4: NMI for K-modes Clustering with Low-level cKML and COS Measures

Data Set	**Intra-cKML**	**Intra-COS**	**Inter-cKML**	**Inter-COS**	**cKML**	**COS**
SoyS	**0.9394**	0.0792	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Zoo	**0.7899**	0.0917	**0.8748**	**0.8748**	0.8403	**0.8708**
DNAP	**0.4363**	0.0095	0.0028	0.0011	**0.5196**	0.0000
Hay	**0.1160**	0.0213	0.0066	0.0136	**0.0284**	0.0209
Lym	**0.1580**	0.1137	0.1527	**0.1714**	**0.1844**	0.1536
Hep	**0.1386**	0.0029	0.1082	**0.1633**	**0.1457**	0.0000
Aud	**0.5107**	0.1475	0.4710	**0.4765**	**0.5283**	0.4601
Hsv	**0.4072**	0.0039	**0.5111**	**0.5111**	**0.5111**	**0.5111**
Spc	**0.0773**	0.0017	0.1008	**0.1035**	**0.1021**	0.0927
Mof	**0.0213**	0.0015	0.0111	**0.0117**	0.0055	**0.0099**
SoyL	**0.7469**	0.1070	**0.6854**	0.6649	**0.6924**	0.6764
Prim	**0.3568**	0.0465	**0.3716**	0.3706	**0.3690**	0.3344
Dmg	**0.7596**	0.0168	0.8358	**0.8510**	**0.8510**	0.8470
Wcs	**0.6833**	0.0024	**0.7973**	0.7216	**0.7944**	0.6881
Crx	**0.0215**	0.0017	**0.0215**	**0.0215**	**0.0215**	0.0013
Br	**0.6675**	0.0024	**0.7486**	0.7117	**0.7835**	0.6597
Ma	**0.2810**	0.0013	0.3227	**0.3250**	**0.3286**	0.2958
Tic	**0.0216**	0.0007	0.0000	0.0000	**0.0265**	0.0029
Flr	**0.2574**	0.0059	**0.5164**	0.3077	0.3171	**0.3575**
Titn	**0.1023**	0.0021	**0.1038**	**0.1038**	0.1038	**0.1039**
DNAN	**0.4648**	0.0004	0.0326	**0.0735**	**0.3921**	0.0580
Spl	**0.5402**	0.0006	0.0708	**0.0737**	**0.3926**	0.0133
Krv	**0.0509**	0.0003	**0.0214**	0.0157	**0.0233**	0.0034
Ld	**0.4951**	0.0029	**0.5072**	0.4533	**0.5094**	0.4303
Ms	**0.2905**	0.0003	**0.4396**	0.4288	0.4209	**0.4396**
Adt	**0.0688**	0.0000	**0.1545**	0.1457	**0.1441**	0.0778
Cnt	**0.0010**	**0.0001**	0.0004	**0.0009**	**0.0026**	0.0001
Mean	**0.3483**	0.0246	**0.3285**	0.3184	**0.3718**	0.3003
AR	**1.0000**	2.0000	1.5185	**1.4815**	**1.2222**	1.7778

4.4.1.2 Coupled Kernel Metric Learning-Enabled *K*-Nearest Neighbors Classification

Incorporating cKML into KNN can demonstrate the flexibility of cKML and its classification performance. The KNN method is chosen because it is among the most popular classification methods. Since KNN is sensitive to the distance or similarity measure, it is a good choice for metric performance comparison. The aforementioned distance and similarity measures are adopted to find the k-nearest neighbor of objects. Subsequently, the classification is done according to these neighbors via KNN.

As discussed in Section 3.4, the data sets are randomly allocate 90% of objects as training data and use the remainder for testing. To reduce noise and randomness, this thesis conducts 20 sampling iterations to partition 20 subsets for each iteration. All experiments are conducted on these 20 partitions. The averaged classification performance is reported in terms of F-score.

Firstly, Hamming and OF metrics are compared with cKML as two baseline measures. The results are shown in Table 4.5, in which the best result is shown in bold face. The proposed cKML achieves an overall performance of 0.7675, which improving by **0.0533** over the best baseline performance of 0.7142 (OF). In most of the data sets, cKML has better performance compared with the baseline methods. In some complex data sets, it has a significant advantage. For example, the F-score improves by **0.1957** in the *Spl* data set and by **0.1429** in the *Ld* data set. In the other data sets, cKML does not achieve the best results; however, it obtains a performance comparable with baseline methods.

The comparisons between cKML and state-of-the-art categorical similarity measures are shown in Table 4.5. The overall performance of cKML is slightly better than the state-of-the-art measures, improving 0.0024 over DILCA. In most cases, the performance of cKML is at the same level as the state-of-the-art measures. However, it is worth noting that cKML dramatically improves KNN classification performance in some complex data sets. For instance, cKML achieves 0.9150, while the best state-of-the-art result from DILCA is only 0.8496, an improvement of **7.7**%.

The performance of the low-level coupling measures is shown in Table 4.6. The overall performance of intra-cKML achieves 0.7545, and that of inter-cKML achieves 0.7649. Meanwhile, intra-COS and inter-COS gain 0.2347 and 0.7649, respectively. The results illustrate intra-cKML is significantly better than intra-COS, while inter-cKML is slightly better than inter-COS. Furthermore, the performance of coupled measure cKML is better than the low-level coupling kernels. On the contrary, the performance of COS is

Table 4.5: F-score for KNN Classification with Different State-of-the-art Similarity Measures

Data Set	cKML	COS	Ahmad	DILCA	Rough	Hamming	OF
SoyS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
Zoo	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9125 ± 0.1468	**1.0000 ± 0.0000**	0.9875 ± 0.0559
DNAP	**0.8821 ± 0.1078**	0.7306 ± 0.1421	0.7714 ± 0.1194	0.8778 ± 0.1136	0.7494 ± 0.1391	0.7390 ± 0.1300	0.7262 ± 0.1291
Hay	0.6692 ± 0.1418	0.8076 ± 0.0921	0.5769 ± 0.1131	0.5758 ± 0.1155	**0.8142 ± 0.0935**	0.6056 ± 0.1198	0.7460 ± 0.1363
Lym	0.8199 ± 0.0943	0.7782 ± 0.1001	**0.8651 ± 0.0690**	0.8432 ± 0.0959	0.8125 ± 0.0821	0.7852 ± 0.0870	0.8071 ± 0.0994
Hep	0.6585 ± 0.1184	0.6405 ± 0.1300	0.6540 ± 0.1325	0.6173 ± 0.1422	0.6401 ± 0.1489	0.6565 ± 0.1519	**0.6619 ± 0.1357**
Aud	0.5546 ± 0.0780	0.4580 ± 0.0882	0.5156 ± 0.1190	**0.6436 ± 0.0379**	0.4017 ± 0.0755	0.5871 ± 0.0946	0.4582 ± 0.0894
Hsv	0.9337 ± 0.0465	0.9428 ± 0.0495	**0.9581 ± 0.0415**	0.9490 ± 0.0414	0.9159 ± 0.0514	0.9266 ± 0.0566	0.9377 ± 0.0530
Spc	0.4955 ± 0.0838	0.5014 ± 0.0904	0.5050 ± 0.0900	0.5053 ± 0.0940	0.5040 ± 0.1064	0.5031 ± 0.0906	**0.5100 ± 0.0962**
Mof	0.7577 ± 0.0914	0.8294 ± 0.0775	0.8294 ± 0.0775	0.7456 ± 0.0821	0.8038 ± 0.0807	0.7524 ± 0.0961	**0.8590 ± 0.0854**
SoyL	**0.9411 ± 0.0504**	0.9345 ± 0.0487	0.9343 ± 0.0495	0.9287 ± 0.0535	0.9005 ± 0.0492	0.8984 ± 0.0721	0.8550 ± 0.0605
Prim	0.2835 ± 0.0949	0.2309 ± 0.0671	0.2830 ± 0.0793	0.2746 ± 0.0756	0.2080 ± 0.0664	**0.2942 ± 0.0953**	0.2185 ± 0.0606
Dmg	0.9742 ± 0.0181	0.9625 ± 0.0281	0.9765 ± 0.0165	**0.9808 ± 0.0202**	0.9276 ± 0.0336	0.9337 ± 0.0350	0.6982 ± 0.0736
Wcs	0.9553 ± 0.0197	0.9390 ± 0.0256	**0.9599 ± 0.0231**	0.9461 ± 0.0243	0.9277 ± 0.0258	0.9498 ± 0.0219	0.9041 ± 0.0289
Crx	0.8278 ± 0.0341	0.7917 ± 0.0388	**0.8294 ± 0.0335**	0.8137 ± 0.0330	0.7769 ± 0.0430	0.7815 ± 0.0441	0.7883 ± 0.0307
Br	0.9589 ± 0.0184	0.9407 ± 0.0284	**0.9634 ± 0.0200**	0.9414 ± 0.0250	0.9265 ± 0.0342	0.9393 ± 0.0233	0.8844 ± 0.0464
Ma	**0.7085 ± 0.0659**	0.7084 ± 0.0631	0.7051 ± 0.0620	0.7039 ± 0.0632	0.7024 ± 0.0633	0.7007 ± 0.0600	0.7049 ± 0.0602
Tic	0.8369 ± 0.0382	0.9056 ± 0.0270	**1.0000 ± 0.0000**	0.8972 ± 0.0379	0.4665 ± 0.0410	0.4656 ± 0.0465	0.7735 ± 0.0335
Flr	0.5438 ± 0.0313	**0.5701 ± 0.0438**	0.5441 ± 0.0339	0.5561 ± 0.0313	0.5588 ± 0.0438	0.5477 ± 0.0423	0.5498 ± 0.0400
Titn	**0.1054 ± 0.0176**	**0.1054 ± 0.0176**	**0.1054 ± 0.0176**	**0.1054 ± 0.0176**	**0.1054 ± 0.0176**	**0.1054 ± 0.0176**	**0.1054 ± 0.0176**
DNAN	0.9135 ± 0.0124	0.7752 ± 0.0121	0.8033 ± 0.0148	**0.9165 ± 0.0139**	0.8146 ± 0.0175	0.7525 ± 0.0123	0.6911 ± 0.0145
Spl	**0.9150 ± 0.0165**	0.7725 ± 0.0219	0.7985 ± 0.0207	0.8496 ± 0.0221	0.8105 ± 0.0181	0.7193 ± 0.0226	0.6929 ± 0.0224
Krv	0.9134 ± 0.0177	0.9177 ± 0.0166	**0.9246 ± 0.0174**	0.9139 ± 0.0205	0.8900 ± 0.0143	0.8505 ± 0.0179	0.9148 ± 0.0168
Ld	**0.6261 ± 0.0209**	0.6211 ± 0.0185	0.6181 ± 0.0198	0.6258 ± 0.0185	0.4789 ± 0.0237	0.4832 ± 0.0221	0.4157 ± 0.0219
Ms	**1.0000 ± 0.0000**	0.9998 ± 0.0006	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
Adt	0.6815 ± 0.0106	0.6813 ± 0.0112	**0.6820 ± 0.0107**	0.6816 ± 0.0114	0.6776 ± 0.0104	0.6800 ± 0.0112	0.6801 ± 0.0104
Mean	**0.7675**	0.7517	0.7617	0.7651	0.7202	0.7176	0.7142
AR	3.0962	3.9808	**2.8462**	3.2885	5.0577	4.9038	4.8269

Table 4.6: F-score for KNN Classification with Low-level cKML and COS Measures

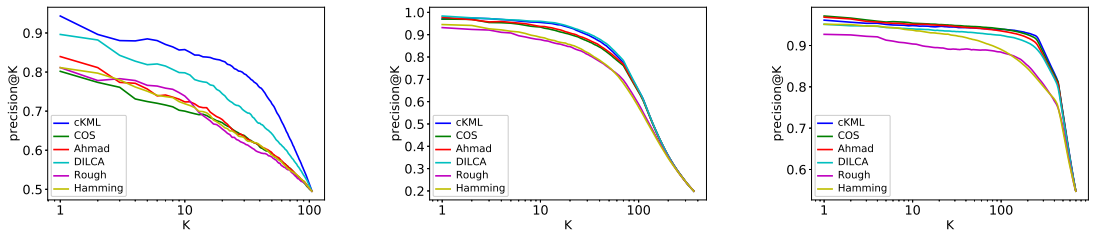
Data Set	Intra-cKML	Intra-COS	Inter-cKML	Inter-COS	cKML	COS
SoyS	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
Zoo	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**
DNAP	**0.9244 ± 0.0641**	0.3333 ± 0.0000	0.7649 ± 0.1179	**0.7714 ± 0.1194**	**0.8821 ± 0.1078**	0.7306 ± 0.1421
Hay	**0.5953 ± 0.1037**	0.1852 ± 0.0000	0.5689 ± 0.1028	**0.5769 ± 0.1131**	0.6692 ± 0.1418	**0.8076 ± 0.0921**
Lym	**0.8336 ± 0.0699**	0.3636 ± 0.0000	0.8629 ± 0.0882	**0.8651 ± 0.0690**	**0.8199 ± 0.0943**	0.7782 ± 0.1001
Hep	**0.6367 ± 0.1178**	0.4444 ± 0.0000	**0.6958 ± 0.1199**	0.6540 ± 0.1325	**0.6585 ± 0.1184**	0.6405 ± 0.1300
Aud	**0.5331 ± 0.0970**	0.0280 ± 0.0089	**0.5400 ± 0.1005**	0.5156 ± 0.1190	**0.5546 ± 0.0780**	0.4580 ± 0.0882
Hsv	**0.8602 ± 0.0629**	0.3041 ± 0.0046	0.9468 ± 0.0446	**0.9581 ± 0.0415**	0.9337 ± 0.0465	**0.9428 ± 0.0495**
Spc	**0.5058 ± 0.0953**	0.4468 ± 0.0000	0.4964 ± 0.0949	**0.5050 ± 0.0900**	0.4955 ± 0.0838	**0.5014 ± 0.0904**
Mof	**0.7050 ± 0.0866**	0.4444 ± 0.0000	0.7955 ± 0.0840	**0.8294 ± 0.0775**	0.7577 ± 0.0914	**0.8294 ± 0.0775**
SoyL	**0.9125 ± 0.0537**	0.0157 ± 0.0000	**0.9347 ± 0.0355**	0.9343 ± 0.0495	**0.9411 ± 0.0504**	0.9345 ± 0.0487
Prim	**0.2857 ± 0.0810**	0.0101 ± 0.0011	**0.2838 ± 0.0750**	0.2830 ± 0.0793	**0.2835 ± 0.0949**	0.2309 ± 0.0671
Dmg	**0.9618 ± 0.0297**	0.0333 ± 0.0000	**0.9809 ± 0.0214**	0.9765 ± 0.0165	**0.9742 ± 0.0181**	0.9625 ± 0.0281
Wcs	**0.9372 ± 0.0280**	0.3982 ± 0.0000	**0.9641 ± 0.0236**	0.9599 ± 0.0231	**0.9553 ± 0.0197**	0.9390 ± 0.0256
Crx	**0.7936 ± 0.0245**	0.3551 ± 0.0000	**0.8331 ± 0.0342**	0.8294 ± 0.0335	**0.8278 ± 0.0341**	0.7917 ± 0.0388
Br	**0.9197 ± 0.0345**	0.3947 ± 0.0000	**0.9666 ± 0.0162**	0.9634 ± 0.0200	**0.9589 ± 0.0184**	0.9407 ± 0.0284
Ma	**0.7070 ± 0.0625**	0.3306 ± 0.0000	**0.7079 ± 0.0605**	0.7051 ± 0.0620	**0.7085 ± 0.0659**	0.7084 ± 0.0631
Tic	**0.7716 ± 0.0307**	0.3949 ± 0.0000	0.9068 ± 0.0263	**1.0000 ± 0.0000**	0.8369 ± 0.0382	**0.9056 ± 0.0270**
Flr	**0.5472 ± 0.0304**	0.0561 ± 0.0020	**0.5597 ± 0.0359**	0.5441 ± 0.0339	0.5438 ± 0.0313	**0.5701 ± 0.0438**
Titn	**0.1054 ± 0.0176**	0.0669 ± 0.0000	**0.1054 ± 0.0176**	**0.1054 ± 0.0176**	**0.1054 ± 0.0176**	**0.1054 ± 0.0176**
DNAN	**0.9287 ± 0.0135**	0.1286 ± 0.0000	**0.8850 ± 0.0190**	0.8033 ± 0.0148	**0.9135 ± 0.0124**	0.7752 ± 0.0121
Spl	**0.9317 ± 0.0119**	0.1310 ± 0.0000	**0.8799 ± 0.0194**	0.7985 ± 0.0207	**0.9150 ± 0.0165**	0.7725 ± 0.0219
Krv	**0.9164 ± 0.0168**	0.3436 ± 0.0000	0.9064 ± 0.0157	**0.9246 ± 0.0174**	0.9134 ± 0.0177	**0.9177 ± 0.0166**
Ld	**0.6263 ± 0.0148**	0.0178 ± 0.0004	**0.6205 ± 0.0194**	0.6181 ± 0.0198	**0.6261 ± 0.0209**	0.6211 ± 0.0185
Ms	**0.9997 ± 0.0009**	0.3817 ± 0.0002	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9998 ± 0.0006
Adt	**0.6802 ± 0.0107**	0.3945 ± 0.0842	**0.6822 ± 0.0111**	0.6820 ± 0.0107	**0.6815 ± 0.0106**	0.6813 ± 0.0112
Mean	**0.7546**	0.3078	**0.7649**	0.7617	**0.7675**	0.7517
AR	**1.0385**	1.9615	**1.3846**	1.6154	**1.3269**	1.6731

even worse than that of inter-COS; this result is consistent with the observations during clustering.

4.4.2 Testing Non-IID Similarity Metrics Learning Quality

4.4.2.1 Coupled Kernel Metric Learning-Enabled Retrieval Performance

What goes further is testing the cKML representation quality in object retrieval, which is a task that heavily depends on data representation. The three data sets DNAP, D-mg, and Br are tested for cKML-enabled retrieval performance evaluation. The results are shown in Figure 4.2, in which the precision and recall of cKML-enabled retrieval outperform that of the others. The results reflect that UNTIE can capture more details of distribution than can other representation methods. The reason lies in UNTIE is powered by hierarchical coupling learning and heterogeneity learning.



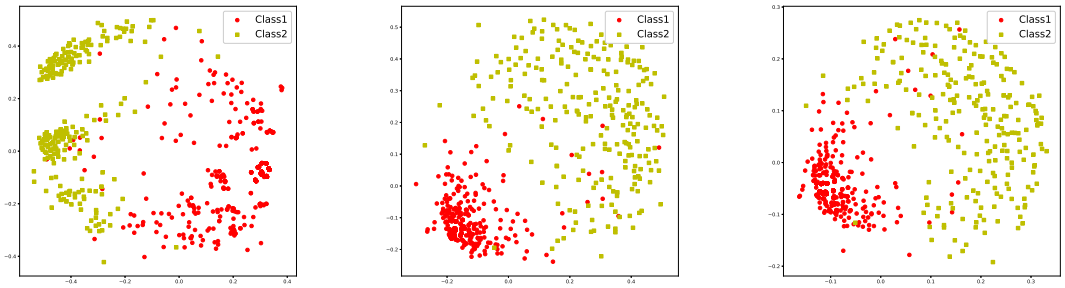
(a) The precision@k-curve on DNAP data set. (b) The precision@k-curve on D-mg data set. (c) The precision@k-curve on Br data set.

(d) The recall@k-curve on DNAP data set. (e) The recall@k-curve on Dmg data set. (f) The recall@k-curve on Br data set.

Figure 4.2: The precision@k-curve and recall@k-curve of different categorical data representation methods: A better metric yields a higher curve in this figure.

4.4.2.2 Coupled Kernel Metric Learning-Represented Data Visualization

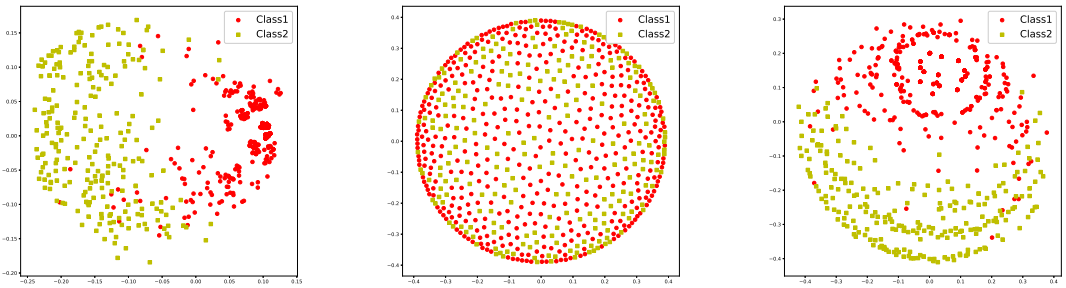
The visualization of cKML-represented data is shown by converting it from the metric space representation to a two-dimensional embedding by multidimensional scaling (Borg & Groenen 2005). Figure 4.3 shows the visualization of different representation methods on Wcs data set. It shows the cKML-represented data has clearer boundaries between different clusters, compared with that of other methods. It qualitatively demonstrates that the cKML-represented data is more suitable for analytics tasks (e.g., classification and clustering).



(a) Distribution in the cKML learned metric space.

(b) Distribution in the COS learned similarity space.

(c) Distribution in the Ahmad learned metric space.



(d) Distribution in the DILCA learned metric space.

(e) Distribution in the Rough learned metric space.

(f) Distribution in the Hamming learned metric space.

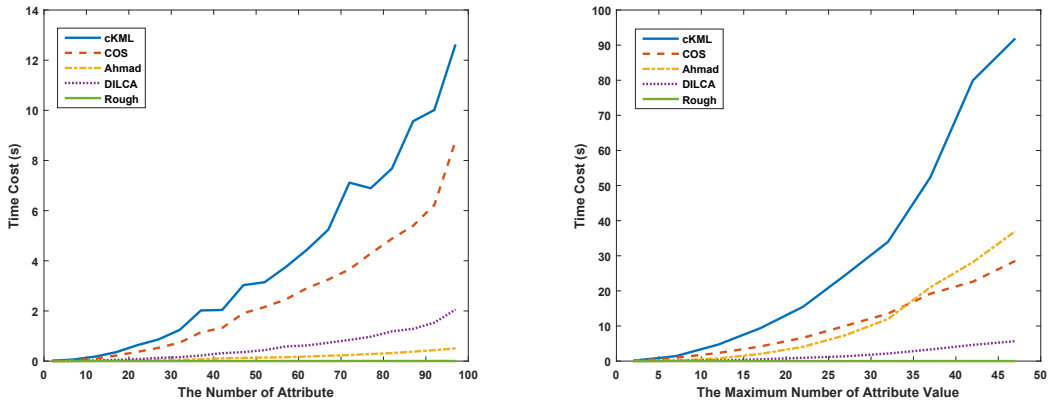
Figure 4.3: The visualization of (dis-)similarity representation of categorical data on Wcs data set. These figures illustrate the data distribution in the cKML learned metric space has clearer boundaries between different clusters. The plotted two-dimensional embedding is converted from the metric space by multidimensional scaling (Borg & Groenen 2005). Different symbols refer to different data clusters per ground truth.

4.4.3 Testing Non-IID Similarity Metrics Learning Efficiency

Synthetic data sets with different data factors are used to empirically analyze the time cost of the proposed cKML, considering the time costs under different settings of the impact of the factors n_a and n_{mv}, respectively.

4.4.3.1 The Impact of Factor n_a

The impact of factor n_a is shown in Figure 4.4(a). In this experiment, synthetic data with 100 objects is generated. The maximum number of values for each attribute (n_{mv}) is fixed at 2, while the number of attributes (n_a) ranges from 2–100. Figure 4.4(a) shows the time cost trend of different categorical similarity measures when factor n_a increases. As shown in the results, all measures except Rough (Cao, Liang, Li, Bai & Dang 2012) are affected by factor n_a. The cKML method has the maximum time cost, while Rough (Cao, Liang, Li, Bai & Dang 2012) has the fastest speed. The time costs of cKML and COS are similar, but cKML has higher cost. This outcome is due to the fact that cKML uses the union value set as context, while COS selects the intersection value set as context for measuring inter-attribute couplings. Although cKML has a slightly higher time cost compared to the other measures, cKML captures more comprehensive coupling relationships in complex data and induces the best learning performance. This result shows some tradeoff between learning performance and computational cost.



(a) Time cost with respect to data factor n_a. (b) Time cost with respect to data factor n_{mv}.

Figure 4.4: Time cost of cKML on synthetic data sets with different data factors.

4.4.3.2 The Impact of Factor n_{mv}

The impact of factor n_{mv} is shown in Figure 4.4(b). In this experiment, the synthetic data also contains 100 generated objects. Here, the number of attributes (n_a) is fixed to 10, and the maximum number of values for each attribute (n_{mv}) ranges from 2 to 50. Figure 4.4(b) shows the computational cost trend of each measure when factor n_{mv} increases, illustrating that cMKL is dramatically affected by factor n_{mv}. This result is consistent with the theoretical analysis of cKML time complexity in Section 4.3.3, which points out that the time cost of cMKL computation has a cubic relationship with factor n_{mv} compared to a square relationship with factor n_a. Compared to the results in Figure 4.4(a), factor n_{mv} generates a higher cost on all measures except Rough (Cao, Liang, Li, Bai & Dang 2012).

4.4.4 Discussion

Theoretical and experimental results show that cKML significantly outperforms the state-of-the-art categorical similarity and dissimilarity measures, and the cKML-enabled learning methods beat their baselines in most of the data sets while achieving comparable performance in other cases. This section discusses on which data characteristics the proposed cKML has superior performance, and it provides a detailed analysis of the intrinsic characteristics that enable cKML to achieve such superiority.

4.4.4.1 Sensitivity to Data Characteristics

First, the sensitivity of cKML to data characteristics is discussed theoretically. The improvement contributed by cKML comes from the complementary information captured by low-level coupling metrics. In other words, cKML is sensitive to the non-IID data with heterogeneity and coupling relationships. The non-IID data complexity can drive cKML to superior performance.

Empirically, cKML shows significant advantage in analyzing DNA sequence data (e.g., *DNAP*, *DNAN*, *Spl*, etc.). The data complexity of this type of data differs from that of others, as there is a clear semantic meaning of DNA in a position. The relationship between different DNA positions also significantly affects the function of the DNA sequence. This non-IID data complexity can be well captured by cKML instead of other state-of-the-art categorical similarity measures, which are designed under the IID assumption. For other types of data, the more complexity it has, the more superior cKML is. Consider a simple index: the difference between the average number of at-

tribute values (n_{av}) and the number of classes (n_c). If n_{av} is much larger than n_c, this size difference reflects that the data is complex for the classification or clustering task. The reason for this complexity is that some attributes belong to the same class in the above case, adding learning complexity due to involving a difference of similarity for the within- and between-class attributes. In this case, cKML may gain better performance compared with other measures. For instance, cKML achieves the best performance on the *Br* data set, where n_{av} is 10 and n_c is 2. As another example, the n_{av} and n_c of the *Wcs* data set are 9.89 and 2, respectively. In this data set, cKML also shows its advantages.

4.4.4.2 Coupled Kernel Metric Learning Intrinsic Nature

The intrinsic nature of cKML is to effectively capture the value-to-object interactions embedded in complex data, represented on the intra- and inter-attribute coupling level, the object coupling level, and the overall coupling.

Regarding intra-attribute coupling, cKML can appropriately capture the value distribution and interaction within an attribute through the well-designed intra-cKML kernel. On the contrary, intra-COS implicitly assumes a specific attribute value distribution (concave distribution), which is not common in reality. Therefore, intra-CKML shows dramatic performance improvement compared with intra-COS.

With respect to inter-attribute coupling, cKML considers attribute interactions under the union co-occurrence value set of given attribute values. This consideration guarantees cKML can comprehensively leverage the co-occurrence information without incurring irrelevant computational cost.

For the object coupling level, cKML jointly seizes the low-level intra- and inter-attribute couplings and combines them to reflect object coupling according to their contribution. As a result, the object coupling involves hierarchical coupling relationships that help to induce superior learning performance.

Moreover, cKML integrates different sources of information using various coupling kernels. This integration guarantees different couplings can be combined together without distortion. In addition, both cKML and COS embed the value similarity space into the value frequency similarity space in order to use the similarity of value frequency to present value similarity. However, cKML adopts the manifold assumption by an exponential mapping to connect these two spaces, while COS uses one space to present the other space directly, which ignores the semantic difference between these spaces.

4.5 Summary

Effectively capturing low-level coupling relationships within and between attributes, labels, and objects is fundamental for appropriate similarity learning, yet challenging due to the intrinsic complexities of modeling the comprehensive and hierarchical couplings in complex data. As shown in Chapter 2, only limited solutions have been proposed for this problem, and the similarity generated by most of them does not satisfy the metric properties, which are required by a large number of downstream learning models. This chapter proposes a coupled metric for categorical data to address this important issue. Compared with existing coupling learning methods, the proposed coupled metric not only comprehensively models value-to-attribute-to-object couplings but also satisfies kernel and distance metric properties. As a result, it achieves significantly better performance in terms of learning intrinsic similarities, clustering, and classification compared to state-of-the-art competitors.

This chapter also shows the significant challenge and importance of deeply understanding the non-IIDness of complex categorical data and, in particular, heterogeneous couplings from different aspects. However, as discussed in Chapter 1, the non-IIDness not only contains multiple couplings studied in this chapter but also includes heterogeneities (e.g., the heterogeneous distributions and dependencies of values, attribute, and objects), which will be discussed in the following chapters.

Part II

Heterogeneity Learning

CHAPTER 5

DECOUPLED NON-IID CATEGORICAL DATA REPRESENTATION

5.1 Introduction

Heterogeneity learning is a critical yet challenging task for non-IID representation learning on complex categorical data. It consists of two primary components: *heterogeneous dependency learning* and *heterogeneous distribution learning*. As analyzed in Section 2.4, the existing heterogeneous distribution learning methods are not available for non-IID categorical data representation. To fill this gap and build heterogeneity learning foundation, this chapter focuses on heterogeneous distribution learning; heterogeneous dependency learning will be discussed in studying non-IID-completeness learning (in Chapters 6 and 7). Different from the similarity-based categorical data representation studied in Chapter 4, this chapter learns a vector-based data representation through a decoupled non-IID learning (DNL) framework. The DNL framework explicitly embeds the non-IIDness of categorical data into a Euclidean space.

In heterogeneous distribution learning on complex data, the biggest obstacle is that attributes are coupled with each other. Because of the coupled attributes, the complexity of data distribution modeling may increase from $n_v^{(1)} + n_v^{(2)} + \cdots + n_v^{(n_a)}$ up to $n_v^{(1)} \times n_v^{(2)} \times \cdots \times n_v^{(n_a)}$ in terms of model parameters, as shown in 2.2.2. This high complexity requires heterogeneous distribution learning method can disentangle attribute

couplings. To address the attribute coupling disentangling problem, DNL decomposes the non-IID categorical data space into several IID subspaces, in which attributes are independent of each other and values within an attribute are identically distributed, and learns the couplings between them. Then, to learn the mixture value distribution, DNL estimates the diverse characteristics of the IID subspaces. Finally, DNL encodes the couplings and heterogeneity as a numerical vector of the categorical data.

Further, DNL is implemented by a nonparametric Bayesian embedding (non-BEND) for categorical data representation. The non-BEND model is built on a Dirichlet multivariate multinomial mixture model. Specifically, non-BEND estimates the complexity of non-IID categorical data by a Dirichlet process and adaptively represents the non-IID categorical data in a vector space with a suitable size according to the data complexity. In addition, this chapter studies how to bound the information lost in the representation process. With theoretical guarantees (see Theorem 5.2), non-BEND can represent non-IID categorical data with bounded information loss. When the amount of data is large enough, non-BEND is a reversible embedding method, which means the original categorical data distribution can be recovered from the embedded numerical representation.

The key contributions made in this chapter include the following:

- *A decoupled non-IID learning framework:* DNL is a novel non-IID categorical data representation framework to capture both couplings and heterogeneities within and between attributes (see details in Section 5.2). It induces a powerful non-IID categorical data representation by embedding complex non-IID characteristics into a numerical Euclidean space composed of IID subspaces.

- *Disentangling attribute couplings:* A Dirichlet multivariate multinomial mixture model non-BEND decomposes a non-IID categorical data space to IID subspaces (as shown in Section 5.3.1), and a collapsed Gibbs sampling algorithm and a variational inference (VI) algorithm are designed for the efficient decomposition (Section 5.3.2). These subspaces reveal the key components of the non-IID space, thus enhancing the representation performance.

- *Learning heterogeneity:* Both heterogeneous distributions and interactions in the decoupled IID spaces are modelled in non-BEND for non-IID categorical data representation (Section 5.3.3), which assures representation to comprehensively capture non-IID characteristics in non-IID data with limited information loss.

- *Bounding the information loss:* A theoretical proof is given about the information loss bound of non-BEND representation (Section 5.4.1). This bound theoretically supports the fact that non-BEND can well represent non-IID categorical data with good generalization.

Comprehensive experiments are conducted on 25 real-life data sets with diversified coupling relationships and heterogeneities and three synthetic data sets generated per a variety of data factors indicating couplings and heterogeneities to evaluate the DNL-based non-BEND model. The results show that, in comparison with five state-of-the-art categorical data representation methods and one-hot embedding and auto-encoder (AE)-based representations, (1) non-BEND learns the non-IID characteristics more effectively; (2) non-BEND achieves a significant accuracy improvement in clustering (averagely 11.75% and maximally 22.37% in terms of F-score) gained from learning complex non-IID characteristics; (3) non-BEND produces generalized representations that are not only suitable for but also enhance the performance of different learning tasks (e.g., on average, 16%, and maximally 3.12%, F-score improvement in classification, and 0.001 root-mean-square error [RMSE] and 0.0008 mean absolute error [MAE] improvement in recommendation); (4) non-BEND-represented data is with good quality that can essentially enhance the downstream learning performance; and (5) the efficiency of non-BEND is insensitive to the volume of data, the number of attributes, and the number of categorical values, which indicates non-BEND is scalable for large amounts of high-dimensional complex non-IID data. The experimental results evidence that the DNL framework and its instance non-BEND are effective and generalizable for non-IID categorical data representation.

The rest of this chapter is organized as follows. Section 5.2 outlines the DNL framework, and Section 5.3 implements DNL by the non-BEND model. Section 5.4 provides the theoretical analysis of the properties of the non-BEND model. Section 5.5 discusses the connections between non-BEND and existing representation methods. Section 5.6 demonstrates the non-BEND performance by comparing it with existing categorical data representation methods for a variety of learning tasks. Lastly, Section 5.7 concludes this chapter.

5.2 The Decoupled Non-IID Learning Framework

Formally, the distributions of categorical data can be represented by a joint distribution of categorical values in each attribute as $\Phi = \{\phi_{v^{(1)}v^{(2)}...v^{(n_a)}} | v^{(j)} \in V^{(j)}\}$, where $V^{(j)}$

refers to the value set of the j-th attribute, $\phi_{v^{(1)}v^{(2)}\ldots v^{(n_a)}}$ is the probability that values $v^{(1)}, v^{(2)}, \cdots, v^{(n_a)}$ co-occur, which follows that $\sum_{v^{(1)}=1}^{n_v^{(1)}} \cdots \sum_{v^{(n_a)}=1}^{n_v^{(n_a)}} \phi_{v^{(1)}v^{(2)}\ldots v^{(n_a)}} = 1$. Here, Φ can be transformed to a tensor $\boldsymbol{\Phi}$, in which each entry equals a value in Φ and is determined by the corresponding categorical values. As analyzed above, for non-IID categorical data, Φ cannot be decomposed as the product of multinomial distributions of each attribute due to the diverse couplings and heterogeneities embedded within and between attributes and then objects. Fortunately, non-IID categorical data distributions can be decomposed as the weighted sum of the product of multinomial distributions in its IID subspaces (Lazarsfeld et al. 1968, Kolda 2001, Dunson & Xing 2009).

Theorem 5.1. *(Dunson & Xing 2009) For any distribution of categorical data, Φ, its corresponding tensor $\boldsymbol{\Phi}$ can be decomposed as $\boldsymbol{\Phi} = \sum_{h=1}^{k} \pi_h \Theta_h$, where $\Theta_h = \boldsymbol{\theta}_h^{(1)} \otimes \boldsymbol{\theta}_h^{(2)} \otimes \cdots \otimes \boldsymbol{\theta}_h^{(n_a)}$, π_h is the weight of Θ_h for composing $\boldsymbol{\Phi}$, $\otimes$ refers to the out product, $\boldsymbol{\theta}_h^{(j)}$ is a probability vector with a size of $n_v^{(j)} \times 1$, for $h = 1, \cdots, k$, and $j = 1, \cdots, n_a$.*

Theorem 5.1 implies that a non-IID data space can be decomposed into several subspaces, where categorical attributes are independent of each other and the values of an attribute are generated from the same distribution (i.e., these subspaces are IID). This theorem provides the foundation for disentangling couplings and transforming heterogeneous distributions to various homogeneous distributions. On this basis, a novel non-IID categorical data representation framework is proposed and implemented in terms of a nonparametric Bayesian approach, as described below.

To tackle the challenges in data representation brought about by non-IID data characteristics, a DNL framework is proposed for categorical data representation. The main idea of the DNL framework is to decompose a complex non-IID space into several simpler IID subspaces. The data characteristics in these IID subspaces and the relations between the IID subspaces are explored in representing the categorical data. Intuitively, this framework simplifies the data complexities and their representation challenges from non-IID to IID and captures both the local characteristics in each subspace and the global structures of the whole space, inducing a powerful categorical data representation.

The DNL framework is illustrated in Figure 5.1. With DNL, the non-IID space of categorical data is decomposed into several IID subspaces by disentangling couplings and heterogeneity. Then the distribution characteristics in each IID subspace and the relations between the non-IID space and the IID subspaces are estimated to construct the embedded vector space for non-IID data representation.

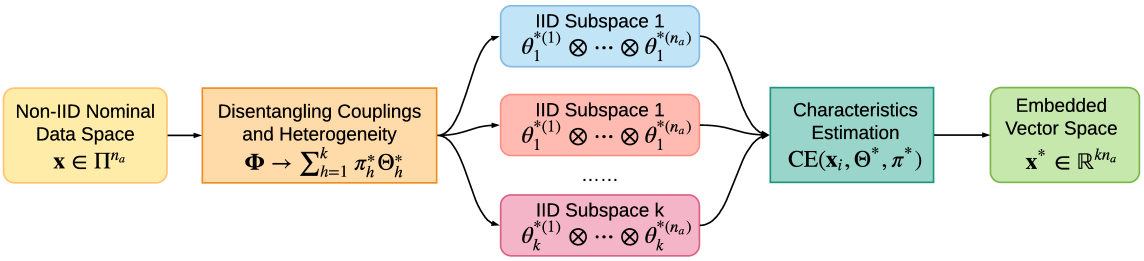


Figure 5.1: Categorical data representation framework with decoupled non-IID learning.

Given a non-IID categorical data set $\mathbf{X} = \{\mathbf{x}_i \in \Pi^{n_a} | 1 \leq i \leq n_o\}$ drawn from Φ, DNL first learns Θ^* and $\boldsymbol{\pi}^*$ by optimizing the following objective function:

$$\begin{aligned} \underset{\Theta^*, \boldsymbol{\pi}^*}{\text{minimize}} \quad & div(\boldsymbol{\Phi} || \textstyle\sum_{h=1}^{k} \pi_h^* \Theta_h^*) \\ \text{subject to} \quad & \Theta_h^* = \boldsymbol{\theta}_h^{*(1)} \otimes \boldsymbol{\theta}_h^{*(2)} \otimes \cdots \otimes \boldsymbol{\theta}_h^{*(n_a)}, \\ & \boldsymbol{\theta}_h^{*(i)} \perp \boldsymbol{\theta}_h^{*(j)}, i, j = 1, \cdots, n_a, i \neq j \end{aligned} \tag{5.1}$$

where $div(\cdot||\cdot)$ is a distribution divergence measurement and $\perp$ denotes the independence of distributions. The objectives of this function are twofold. First, it learns subspaces with value distributions Θ^* which can approximate the original non-IID space with bounded information loss in terms of distribution divergence. Second, attributes in these learned subspaces are independent of each other, and each attribute has a homogeneous distribution $\boldsymbol{\theta}_h^{*(j)}$. These two parts serve the purpose of disentangling the couplings and heterogeneities in the non-IID space and decomposing the space into several IID parts. In this way, DNL captures the statistical nature of couplings and heterogeneities. Further, for each subspace, DNL estimates its characteristics and uses an estimator function $CE(\cdot,\cdot,\cdot)$ to embed the categorical data into a Euclidean space. The estimator function considers both local information involved in Θ^*, which reflects the distributions of each subspace, and the global information reflected by $\boldsymbol{\pi}^*$, which reveals the interactions between the original non-IID space and these subspaces. Formally, the embedded data set $\mathbf{X}^*$ can be obtained by:

$$\mathbf{X}^* = \{CE(\mathbf{x}_i, \Theta^*, \boldsymbol{\pi}^*) | 1 \leq i \leq n_o\}. \tag{5.2}$$

The DNL framework can be flexibly instantiated into a variety of specific models by adopting different strategies in each step. For example, the Kullback–Leibler divergence

with the gradient descent method and the non-negative tensor factorization method can be used in the first step, and the principal component analysis and the autoencoder methods can be used in the second step. Prior and domain knowledge can also be added into DNL to better model the structure of non-IID space and obtain a more accurate estimator of data characteristics.

For the different strategies adopted in the implementation, DNL offers the essential representation capability of non-IID categorical data: (1) It decouples the interactions between attributes, which provides fundamental bases and intrinsic descriptions of coupled categorical data; and (2) it learns the heterogeneities in multiple decomposed IID subspaces, which is a simple but effective way to build the whole picture of the non-IID space. The thesis instantiates DNL by a nonparametric Bayesian embedding method, which naturally models the non-IID space decomposition and estimates data characteristics, to demonstrate the insight and performance of the proposed framework.

5.3 Non-BEND: DNL-Based Nonparametric Bayesian Embedding

This section introduces a nonparametric Bayesian embedding method, non-BEND, to implement DNL for categorical data representation. The non-BEND model adopts a Dirichlet multivariate multinomial mixture model to disentangle couplings and heterogeneities. It further uses the expectation of the distribution in each IID subspace to represent categorical data characteristics. The assumption of the prior distribution of non-IID categorical data is first given, followed by the posterior inference, and the embedding algorithm is then detailed.

5.3.1 The Prior Distribution of Non-IID Categorical Data

The generative process of non-IID categorical data representation by non-BEND is given as follows.

(1). Draw the mixed parameters of the Dirichlet multivariate multinomial mixture from a Dirichlet process. For this step, first draw the subspace assignment z_i for each object $\mathbf{x}_i$ from a Chinese restaurant process (CRP):

$$z_i \sim CRP(\alpha). \tag{5.3}$$

Then, draw the parameters of multinomial distribution for each attribute in each subspace from the Dirichlet distribution:

$$\boldsymbol{\theta}_{z_i}^{(j)} \sim Dirichlet(\boldsymbol{\alpha}_0), \tag{5.4}$$

where α and $\boldsymbol{\alpha}_0$ are the concentration parameters for CRP and the Dirichlet distribution, respectively.

(2). Draw the attribute value in the j-th attribute $x_i^{(j)}$ of object $\mathbf{x}_i$ from the multinomial distribution with parameters $\boldsymbol{\theta}_{z_i}^{(j)}$:

$$x_i^{(j)} \sim multinomial(\boldsymbol{\theta}_{z_i}^{(j)}), \tag{5.5}$$

for $i = 1, \cdots, n_o$ and $j = 1, \cdots, n_a$. Figure 5.2 shows the graphical model of the generative process.

The inference of this process is equivalent to solving Equation (5.1) by assuming π_h^* to be the occurrence probability of the h-th subspace. This equivalence holds because the gap between the non-IID categorical data distribution Φ and the estimated $\sum_{h=1}^{k} \pi_h^* \Theta_h^*$ approaches zero when the amount of data increases, which is guaranteed by Theorem 5.2, as shown in the theoretical analysis (see Section 5.4).

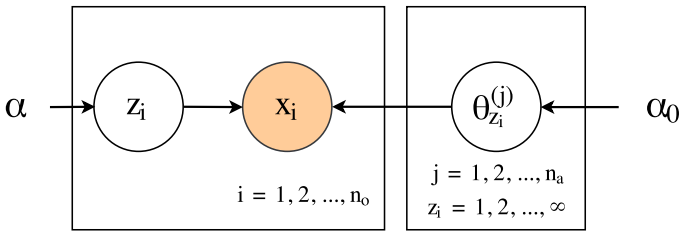


Figure 5.2: The graphical model of non-BEND.

5.3.2 The Posterior Probability of Non-IID Categorical Data

5.3.2.1 Gibbs Sampling

This section provides the Gibbs sampling to estimate the parameters of the generative process. Sampling the subspace assignment of each object requires the posterior probability of $\mathbf{x}_i$ belonging to the subspace h, given other sampled parameters, $P(z_i = h|\mathbf{X}, \mathbf{z}_{-i}, \alpha, \boldsymbol{\alpha}_0, \boldsymbol{\theta})$. Here, $\mathbf{z}_{-i}$ refers to the assignments of subspaces excluding the

i-th object. This posterior probability can be decomposed into three components, as follows.

$$
\begin{aligned}
&P(z_i = h|\mathbf{X}, \mathbf{z}_{-i}, \alpha, \boldsymbol{\alpha}_0, \boldsymbol{\theta}) \\
&= P(z_i = h|\mathbf{z}_{-i}, \alpha) \cdot P(\mathbf{z}_{-i}, \alpha|\mathbf{X}, \boldsymbol{\alpha}_0, \boldsymbol{\theta}) \\
&= P(z_i = h|\mathbf{z}_{-i}, \alpha) \cdot P(\mathbf{z}_{-i}, \alpha|\mathbf{X}) \cdot P(\mathbf{X}|\boldsymbol{\alpha}_0, \boldsymbol{\theta}) \\
&= P(z_i = h|\mathbf{z}_{-i}, \alpha) \cdot P(\mathbf{z}_{-i}, \alpha|\mathbf{X}) \cdot \prod_{j=1}^{n_o} P(\mathbf{x}_j|\boldsymbol{\alpha}_0, \boldsymbol{\theta}).
\end{aligned} \tag{5.6}
$$

For the z_i's assignment, the component $P(\mathbf{z}_{-i}, \alpha|\mathbf{X})$ and the conditional probability of $\mathbf{X}_{-i}$, which refers to the objects excluding $\mathbf{x}_i$, in the component $\prod_{j=1, j\neq i}^{n_o} P(\mathbf{x}_j|\boldsymbol{\alpha}_0, \boldsymbol{\theta})$, are constant values. Equation (5.6) can thus be reduced as follows:

$$
P(z_i = h|\mathbf{X}, \mathbf{z}_{-i}, \alpha, \boldsymbol{\alpha}_0, \boldsymbol{\theta}) \propto P(z_i = h|\mathbf{z}_{-i}, \alpha) \cdot P(\mathbf{x}_i|\boldsymbol{\alpha}_0, \boldsymbol{\theta}). \tag{5.7}
$$

The component $P(z_i = h|\mathbf{z}_{-i}, \alpha)$ in Equation (5.7) is the probability of the subspace assignment of an object conditional on the subspace assignment of other objects and the concentration parameter α. Since the subspace assignments were drawn from the CRP, this component equals the probability of an object joining an existing table when h has already appeared, or alternatively, the probability of joining a new table. These probabilities are formally given in Equation (5.8):

$$
P(z_i = h|\mathbf{z}_{-i}, \alpha) = \begin{cases} \frac{t_h^{\mathbf{z}_{-i}}}{n_o - 1 + \alpha} & \text{if } h \leq \ell \\ \frac{\alpha}{n_o - 1 + \alpha} & \text{if } h = \ell + 1, \end{cases} \tag{5.8}
$$

where $t_h^{\mathbf{z}_{-i}}$ refers to the number of elements in $\mathbf{z}_{-i}$, which belongs to the subspace h, and where ℓ is the number of currently occupied subspaces.

The component $P(\mathbf{x}_i|\boldsymbol{\alpha}_0, \boldsymbol{\theta})$ may also have different values. When z_i belongs to an existing subspace, $\boldsymbol{\theta}$ is a constant. It can be collapsed to increase the sampling speed, since non-BEND only needs to concern the IID subspace assignment for disentangling couplings and heterogeneities in non-IID categorical data. After the collapse of $\boldsymbol{\theta}$, the probability of $\mathbf{x}_i$ becomes conditional on an existing subspace h and the concentration parameters $\boldsymbol{\alpha}_0$. Because the Dirichlet distribution is the conjugate prior of a multinomial distribution, the conditional probability of $\mathbf{x}_i$ actually equals the posterior probability

of the Dirichlet distribution for the given observations $\mathbf{X}_{-i}$, that is,

$$P(\mathbf{x}_i|\boldsymbol{\alpha}_0,\boldsymbol{\theta}) = \frac{C(\mathbf{u}_{z_i}^{\mathbf{x}_i^{(1)}} + \mathbf{u}_{z_i}^{\mathbf{X}_{-i}^{(1)}} + \boldsymbol{\alpha}_0^{(1)})}{C(\mathbf{u}_{z_i}^{\mathbf{X}_{-i}^{(1)}} + \boldsymbol{\alpha}_0^{(1)})} \cdots \frac{C(\mathbf{u}_{z_i}^{\mathbf{x}_i^{(n_a)}} + \mathbf{u}_{z_i}^{\mathbf{X}_{-i}^{(n_a)}} + \boldsymbol{\alpha}_0^{(n_a)})}{C(\mathbf{u}_{z_i}^{\mathbf{X}_{-i}^{(n_a)}} + \boldsymbol{\alpha}_0^{(n_a)})}, \tag{5.9}$$

where

$$\begin{aligned} C(\boldsymbol{\beta}^{(i)}) &= \int \prod_{j=1}^{n_v^{(i)}} (\theta_j^{(i)})^{\beta_j^{(i)}-1} \mathrm{d}\boldsymbol{\theta}^{(i)} \\ &= \prod_{j=1}^{n_v^{(i)}} \Gamma(\beta_j^{(i)}) / \Gamma(\sum_{j=1}^{n_v^{(i)}} \beta_j^{(i)}), \end{aligned}$$

and where $\mathbf{u}_{z_i}^{\mathbf{X}^{(j)}} = [u_{v_1^{(j)}}^{\hat{\mathbf{X}}^{(j)}}, \cdots, u_{v_{n_v^{(j)}}^{(j)}}^{\hat{\mathbf{X}}^{(j)}}]$ is a vector that counts the number of values in the j-th attribute appearing in the data set $\hat{\mathbf{X}} \subseteq \mathbf{X}$, which belongs to the subspace z_i. When z_i is assigned as a new subspace, the $\boldsymbol{\theta}$ of the new subspace should be drawn from a Dirichlet process based on $\boldsymbol{\alpha}_0$. In this case, $\mathbf{x}_i$ is drawn from a Dirichlet multivariate multinomial mixture, which has the following probability:

$$\begin{aligned} P(\mathbf{x}_i|\boldsymbol{\alpha}_0,\boldsymbol{\theta}) &= \int \cdots \int P(\mathbf{x}_i^{(1)}|\boldsymbol{\theta}^{(1)}) \cdots P(\mathbf{x}_i^{(n_a)}|\boldsymbol{\theta}^{(n_a)}) \cdot \\ &\quad P(\boldsymbol{\theta}^{(1)}|\boldsymbol{\alpha}_0^{(1)}) \cdots P(\boldsymbol{\theta}^{(n_a)}|\boldsymbol{\alpha}_0^{(n_a)}) \mathrm{d}\boldsymbol{\theta}^{(1)} \cdots \mathrm{d}\boldsymbol{\theta}^{(n_a)} \\ &= \int \cdots \int \prod_{j=1}^{n_v^{(1)}} (\theta_j^{(1)})^{u_{v_j^{(1)}}^{\mathbf{x}_i^{(1)}}} \cdots \prod_{j=1}^{n_v^{(n_a)}} (\theta_j^{(n_a)})^{u_{v_j^{(n_a)}}^{\mathbf{x}_i^{(n_a)}}} \cdot \\ &\quad \prod_{j=1}^{n_v^{(1)}} (\theta_j^{(1)})^{\alpha_{0j}^{(1)}-1} \cdots \prod_{j=1}^{n_v^{(n_a)}} (\theta_j^{(n_a)})^{\alpha_{0j}^{(n_a)}-1} \cdot \\ &\quad \frac{1}{C(\boldsymbol{\alpha}_0^{(1)})} \cdots \frac{1}{C(\boldsymbol{\alpha}_0^{(n_a)})} \mathrm{d}\boldsymbol{\theta}^{(1)} \cdots \mathrm{d}\boldsymbol{\theta}^{(n_a)} \\ &= \frac{C(\mathbf{u}_{z_i}^{\mathbf{x}_i^{(1)}} + \boldsymbol{\alpha}_0^{(1)})}{C(\boldsymbol{\alpha}_0^{(1)})} \cdots \frac{C(\mathbf{u}_{z_i}^{\mathbf{x}_i^{(n_a)}} + \boldsymbol{\alpha}_0^{(n_a)})}{C(\boldsymbol{\alpha}_0^{(n_a)})}. \end{aligned} \tag{5.10}$$

By combining Equations (5.8), (5.9), and (5.10) with Equation (5.7), the posterior

probability of the subspace assignment of each object can be obtained as follows:

$$
\begin{aligned}
&P(z_i = h|\mathbf{x}, \mathbf{z}_{-i}, \alpha, \boldsymbol{\alpha}_0, \boldsymbol{\theta}) \propto \\
&\begin{cases}
\dfrac{t_h^{(\mathbf{z}_{-i})}}{n_o - 1 + \alpha} \cdot \dfrac{C(\mathbf{t}_{z_i}^{\mathbf{x}_i^{(1)}} + \mathbf{t}_{z_i}^{\mathbf{x}_{-i}^{(1)}} + \boldsymbol{\alpha}_0^{(1)})}{C(\mathbf{t}_{z_i}^{\mathbf{x}_{-i}^{(1)}} + \boldsymbol{\alpha}_0^{(1)})} \cdots \dfrac{C(\mathbf{t}_{z_i}^{\mathbf{x}_i^{(n_a)}} + \mathbf{t}_{z_i}^{\mathbf{x}_{-i}^{(n_a)}} + \boldsymbol{\alpha}_0^{(n_a)})}{C(\mathbf{t}_{z_i}^{\mathbf{x}_{-i}^{(n_a)}} + \boldsymbol{\alpha}_0^{(n_a)})} \\
\text{if } h \le \ell \\
\dfrac{\alpha}{n_o - 1 + \alpha} \cdot \dfrac{C(\mathbf{t}_{z_i}^{\mathbf{x}_i^{(1)}} + \boldsymbol{\alpha}_0^{(1)})}{C(\boldsymbol{\alpha}_0^{(1)})} \cdots \dfrac{C(\mathbf{t}_{z_i}^{\mathbf{x}_i^{(n_a)}} + \boldsymbol{\alpha}_0^{(n_a)})}{C(\boldsymbol{\alpha}_0^{(n_a)})} \\
\text{if } h = \ell + 1.
\end{cases}
\end{aligned} \tag{5.11}
$$

5.3.2.2 Variational Inference

This section provides a VI method to estimate the non-BEND parameters for a more efficient representation of large non-IID categorical data.

To achieve the more efficient VI, the stick-breaking process is used to take the place of CRP in the non-BEND generative process. In this way, the subspace assignment z_i for each object $\mathbf{x}_i$ is drawn from the stick-breaking process. Denoting the stick lengths as $\boldsymbol{\pi} = \{\pi_1, \cdots, \pi_\infty\}$, in which π_k refers the length of the k-th stick, the mixed parameters of the Dirichlet multivariate multinomial mixture in non-BEND can be drawn as follows.

(1). For the h-th stick, the stick-breaking is constructed as follows:

$$v_h \sim Beta(1, \alpha), \quad h = \{1, 2, ..., \infty\}; \tag{5.12}$$

$$\pi_h(\mathbf{v}) = v_h \prod_{l=1}^{h-1} (1 - v_l) \tag{5.13}$$

where the stick lengths $\pi_h(\mathbf{v})$ are given by repeatedly breaking a stick of initial length 1 at points given by the v_h.

(2). For the i-th object, its subspace assignment z_i is drawn according to a multinomial distribution with the stick lengths $\boldsymbol{\pi}$ as parameters:

$$z_i \sim multinomial(\boldsymbol{\pi}). \tag{5.14}$$

(3). For the j-th attribute of the objects in the subspace z_i, its value distribution parameters $\boldsymbol{\theta}_{z_i}^{(j)}$ are drawn from a Dirichlet distribution with the concentration parameters $\boldsymbol{\alpha}_0$:

$$\boldsymbol{\theta}_{z_i}^{(j)} \sim Dirichlet(\boldsymbol{\alpha_0}). \tag{5.15}$$

Subsequently, the inference problem is converted into an optimization problem by the VI method. For representation convenience, the section denotes the target distribution as p and another simple and well-known distribution as q. Variational inference learns the distribution q to mimic the target distribution p, which is always achieved by minimizing the KL divergence between p and q. Because it is very difficult to directly minimize the KL-divergence between p and q, VI alternatively optimizes its equivalent objective (i.e., maximizing the evidence lower bound [ELBO]). The ELBO of the non-BEND model can be derived as follows:

$$\begin{aligned}\text{ELBO} &= E_q[\log p(\mathbf{V}|\alpha)] + E_q[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha}_0] \\ &+ \sum_{i=1}^{n_0}(E_q[\log p(Z_i|\mathbf{V})]) + \sum_{j=1}^{n_a} E_q[\log p(x_i^{(j)}|Z_i)]) \\ &- E_q[\log q(\mathbf{V},\boldsymbol{\theta},\mathbf{Z})].\end{aligned} \tag{5.16}$$

The VI method defines a fully factorized mean-field variational distribution to mimic the prior categorical data distribution as follows:

$$q(\mathbf{V},\boldsymbol{\theta},\mathbf{Z}) = \prod_{t=1}^{n_t-1} q_{\gamma_t}(v_t)\prod_{t=1}^{n_t}\prod_{j=1}^{n_a} q_{\tau_t^{(j)}}(\theta_t^{(j)})\prod_{i=1}^{n_0} q_{\phi_i}(z_i), \tag{5.17}$$

where n_t refers to a truncation level to mimic the steak-breaking process. As was the case for Blei et al. (2006), this thesis fixes n_t and constrains $q(v_{n_t} = 1) = 1$, so that the mixture proportions (namely stick lengths) $\pi_t(\mathbf{v})$ for $t > n_t$ are equal to zero with probability 1. For a better approximation result, all the variational distributions (i.e., $q_\cdot(\cdot)$ in equations) have the same forms but parameters that differ from the target distributions (i.e., $p(\cdot)$ in equations). A coordinate ascent algorithm is used for optimizing the ELBO in Equation (5.16). Furthermore, this thesis follows with the general expression in Blei et al. (2006) and use Dirichlet distribution as the base measure. For $t \in \{1,\cdots,n_t\}, i \in \{1,\cdots,n_o\}, j \in \{1,\cdots,n_a\}$, the updating of each variational parameter is as follows:

$$\gamma_{t,1} = 1 + \sum_{i=1}^{n_o}\phi_{i,t}, \tag{5.18}$$

$$\gamma_{t,2} = \alpha + \sum_{i=1}^{n_o}\sum_{l=t+1}^{n_t}\phi_{i,l}, \tag{5.19}$$

$$\tau_t^{(j)} = \alpha_0 + \sum_{i=1}^{n_o}\phi_{i,t}u^{\hat{\mathbf{X}}^{(j)}}_{x_i^{(j)}}, \tag{5.20}$$

$$\phi_{i,t} \propto exp(S_{i,t}), \tag{5.21}$$

and

$$S_{i,t} = E_q[\log V_t] + \sum_{l=1}^{n_t-1} E_q[\log(1-V_l)] + \sum_{j=1}^{n_a} E_q[\log\theta^{(j)}_{x_i^{(j)},t}] u^{\hat{\mathbf{x}}^{(j)}}_{x_i^{(j)}}, \tag{5.22}$$

where

$$E_q[\log V_t] = \psi(\gamma_{t,1}) - \psi(\gamma_{t,1}+\gamma_{t,2}) \tag{5.23}$$

and

$$E_q[\log(1-V_t)] = \psi(\gamma_{t,2}) - \psi(\gamma_{t,1}+\gamma_{t,2}) \tag{5.24}$$

and where $\psi(\cdot)$ is the *digamma* function, which appears when taking the derivative of the log normalizer in the beta distribution. For each attribute j,

$$E_q[\log\theta^{(j)}_{x_i^{(j)},t}] = \psi(\tau^{(j)}_{x_i^{(j)},t}) - \psi(\sum_{j=1}^{n_a} \tau^{(j)}_{x_i^{(j)},t}). \tag{5.25}$$

5.3.3 The Non-BEND Representation

The non-BEND model represents non-IID categorical data following three steps. In the first step, non-BEND decomposes non-IID categorical data to IID subspaces. In the second step, non-BEND estimates data characteristics in each IID subspace. Finally, in the third step, non-BEND embeds the estimated characteristics in the categorical data representation.

The first step can be implemented either by the proposed collapsed Gibbs sampling or VI. The Algorithm 1 summarizes the process of the non-BEND model based on the collapsed Gibbs sampling. First, the parameters of the generative process of non-BEND are estimated by the collapsed Gibbs sampling, as shown in lines (2)–(5). Subsequently, non-BEND assigns the subspace to each object according to the mode of their assigned subspaces after the burn-in stage. Similarly, Algorithm 2 shows the representation process of the non-BEND model based on the VI, in which the coordinate ascent algorithm is used for the ELBO optimization.

After decoupling the non-IID space, the second step of non-BEND estimates the multinomial distribution of the j-th attribute in the h-th subspace as follows:

$$\boldsymbol{\theta}^{*(j)}_h = E(\boldsymbol{\theta}^{(j)}_h) = \left(\frac{\alpha^{(j)}_{01} + m^{(j)}_{h1}}{\sum_{i=1}^{n^{(j)}_v} \alpha^{(j)}_{0i} + m_h}, \cdots, \frac{\alpha^{(j)}_{0n^{(j)}_v} + m^{(j)}_{hn^{(j)}_v}}{\sum_{i=1}^{n^{(j)}_v} \alpha^{(j)}_{0i} + m_h} \right), \tag{5.26}$$

Algorithm 1 The collapsed Gibbs sampling for non-BEND

Require: The concentration parameters $\boldsymbol{\alpha}_0$ and α, the categorical data $\mathbf{X}$, the Gibbs sampling epochs $maxEpoch$, and the burn-in length $burnin$.

Ensure: The embedded numerical data $\mathbf{X}^*$.

Initialization by randomly assigning each object $\mathbf{x}_i$ to a subspace z_i, $epoch = 0$
for $epoch < maxEpoch$ **do**
    Sampling z_i from Equation (5.11), for $i = 1, \cdots, n_o$.
    $epoch = epoch + 1$.
end for
$z_i^* = mode(z_i^{(burin+1:maxEpoch)})$.
$\boldsymbol{\theta}_{z_i}^{*(j)} = E(\boldsymbol{\theta}_{z_i}^{(j)})$ as Equation (5.26)
Embedding $\mathbf{X}^*$ per Equation (5.32).
return $\mathbf{X}^*$

Algorithm 2 The mean-field VI for non-BEND

Require: The concentration parameters $\boldsymbol{\alpha}_0$ and α, the categorical data $\mathbf{X}$, the truncation level n_t.

Ensure: The embedded numerical data $\mathbf{X}^*$.

Initialize the variational parameters $\{\gamma_{t,1}, \gamma_{t,2}, \tau_t^{(j)}, \phi_{i,t}\}$.
repeat
    for $j \in \{1, 2, \cdots, n_a\}$ **do**
        for $t \in \{1, 2, \cdots, n_t\}$ **do**
            Update $\gamma_{t,1}$ and $\gamma_{t,2}$, the variational parameters of $\mathbf{V}$ using Equation (5.18) and Equation (5.19)
            Update $\tau_t^{(j)}$, the variational parameters of $\boldsymbol{\theta}$ by using Equation (5.20)
            for $i \in \{1, 2, \cdots, n_o\}$ **do**
                Update $\phi_{i,t}$, the variational parameters of $\mathbf{Z}$ by using Equation (5.21) and Equation (5.22)
            end for
        end for
    end for
until convergence
for $i \in \{1, 2, \cdots, n_o\}$ **do**
    $z_i^* = \underset{t}{\operatorname{argmax}}(\phi_{i,t})$
end for
$\boldsymbol{\theta}_{z_i^*}^{*(j)} = E(\boldsymbol{\tau}_{z_i^*}^{(j)})$ as Equation (5.26)
Embedding $\mathbf{X}^*$ per Equation (5.32)
return $\mathbf{X}^*$

which equals the expectation of the Dirichlet distribution posterior. In Equation (5.26), $\alpha_{0i}^{(j)}$ refers to the i-th prior pseudo-count of the Dirichlet distribution, $m_{hi}^{(j)}$ refers to the number of objects in the h-th subspace with the i-th categorical value, and m_h refers to the number of objects in the h-th subspace. Meanwhile, non-BEND estimates the weight π_h of the h-th IID subspace as the occurrence frequency in the non-IID space:

$$\pi_h^* = \frac{|\{\mathbf{x}_i | z_i = h\}|}{n_o}. \tag{5.27}$$

In the third step, non-BEND represents the categorical value in the j-th attribute of data $\mathbf{x}_i$ per the h-th IID subspace as follows:

$$\mathbf{x}_{i,h}^{*(j)} = \exp(-\theta_{h,x_i^{(j)}}^{*(j)}) \cdot OZ(\mathbf{x}_i^{(j)}), \tag{5.28}$$

where $\exp(\cdot)$ refers to the exponential of the mathematical constant and where $OZ(\cdot)$ refers to the *one-zero* embedding. The *one-zero* embedding represents the t-th unique categorical value in the j-th attribute as a $n_v^{(j)}$-dimensional vector where only the t-th entry is 0 and where all other entries are 1. For example, if an attribute has three unique categorical values $\{A, B, C\}$, one-zero embedding will represent these values as $OZ(A) = [0,1,1]$, $OZ(B) = [1,0,1]$, and $OZ(C) = [1,1,0]$, respectively. In Equation (5.28), the first term $\exp(-\theta_{h,x_i^{(j)}}^{*(j)})$ captures the value distribution in the h-th IID subspace, and the second term $OZ(\mathbf{x}_i^{(j)})$ differentiates categorical values. This step jointly considers these two terms for two reasons. The first reason is that the one-zero embedding can differentiate categorical values but cannot specify the detailed difference between them. In other words, with one-zero embedding, the Euclidean distances between all different categorical values are the same. Therefore, it is necessary to integrate one-zero embedding with other information to enrich the differences between different categorical values. The second reason is that embedding the value distribution in IID subspaces can effectively reveal sufficient information contained in a non-IID space, as proved in Theorem 5.2, but it cannot differentiate categorical values under Euclidean distance when these values have very similar occurrence probabilities. Since non-BEND targets on learning representation in a Euclidean space, it is necessary to complement the estimated IID space characteristics by one-zero embedding. Non-BEND further considers the interactions between different IID subspaces to represent the categorical value $\mathbf{x}_i^{(j)}$ per all IID subspaces as follows:

$$\mathbf{x}_i^{*(j)} = \left[\pi_1^* \mathbf{x}_{i,1}^{*(j)}, \pi_2^* \mathbf{x}_{i,2}^{*(j)}, \cdots, \pi_k^* \mathbf{x}_{i,k}^{*(j)}\right]. \tag{5.29}$$

Finally, as attributes are independent in each IID subspace, non-BEND concatenates the representations of categorical values in each attribute of categorical data $\mathbf{x}_i$ as its representation:

$$\mathbf{x}_i^* = \left[\mathbf{x}_i^{*(1)}, \mathbf{x}_i^{*(2)}, \cdots, \mathbf{x}_i^{*(n_a)}\right]. \tag{5.30}$$

The non-BEND adopts Equation (5.32) to represent categorical data that jointly embeds: (1) the distribution characteristics reflected by Θ^* in IID subspaces, (2) the interactions between different IID subspaces reflected by $\boldsymbol{\pi}^*$, and (3) the essential differences between categorical values. Although the non-BEND-embedded vector representation comprehensively reflects categorical characteristics in a Euclidean space, it ignores different amount of informativeness contained in different attributes. If some attributes do not contain as much information as others, the values in these attributes may have very similar distribution in some IID subspaces. As a result, the Euclidean distance conducted on the above representation may over-consider the information contained in these attributes, because the representations of values in these attributes per different IID subspaces will be similar. To further leverage the significance of different attributes, the representation of the j-th attributes are weighted as follows:

$$w_j = \frac{1}{n_v^{(j)}} \sum_{t=1}^{n_v^{(j)}} \sqrt{\frac{1}{k-1} \sum_{h=1}^{k-1} \left(\theta^{*(j)}_{h,v_t^{(j)}} - \frac{1}{k} \sum_{h=1}^{k} \theta^{*(j)}_{h,v_t^{(j)}} \right)^2}, \tag{5.31}$$

which is the mean value of the standard deviation of the estimated probability of all values in all IID subspaces. With this attribute weighting strategy, the non-BEND representation can be re-formalized as follows:

$$\mathbf{x}_i^* = \left[w_1\mathbf{x}_i^{*(1)}, w_2\mathbf{x}_i^{*(2)}, \cdots, w_{n_a}\mathbf{x}_i^{*(n_a)}\right]. \tag{5.32}$$

5.4 Theoretical Analysis of Non-BEND Properties

This section analyzes the non-BEND effectiveness and efficiency in terms of information loss bound and computational complexity, respectively.

5.4.1 Information Loss Bound of Non-BEND

The non-BEND representation does not lose any information when the amount of data increases. The loss bound is given in Theorem 5.2. Before proving Theorem 5.2, this section introduces Lemma 5.1 and Lemma 5.2.

Lemma 5.1. *Denote the actual probability tensor of categorical data as* Φ^0 *and the probability tensor reconstructed by the non-BEND representation as* Φ^**; let* $\mathcal{N}_\epsilon(\Phi^0) = \{\Phi : \|\Phi - \Phi^0\|_1 < \epsilon\}$ *denote an* L_1 *neighbor around* Φ^0*, for any* $\epsilon > 0$*,* $\lim_{n_o \to \infty} Pr\{\Phi^* \in \mathcal{N}_\epsilon(\Phi^0)|\mathbf{X}\} \to 1$*, where* $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ *is the data for all* n_o *objects.*

Proof. Because a Dirichlet process-based multivariate multinomial mixture is used as the prior of categorical data, according to Theorem 2 in Dunson & Xing (2009), the probability $P\{\mathcal{N}_\epsilon(\Phi^0)\} > 0$ for any $\epsilon > 0$. Since Φ^* includes finite parameters, it is sufficient for posterior consistency, which guarantees $\lim_{n_o \to \infty} Pr\{\Phi^* \in \mathcal{N}_\epsilon(\Phi^0)|\mathbf{X}\} \to 1$ for any $\epsilon > 0$. ■

Lemma 5.2. *Given a non-IID categorical data set* $\mathbf{X}$ *drawn from* Φ*, let the non-BEND-learned local and global parameters be* Θ^* *and* $\boldsymbol{\pi}^*$*, respectively, for any f-divergence* $D_f(\cdot||\cdot)$*,* $\lim_{n_o \to \infty} D_f(\Phi||\sum_{h=1}^{k} \pi_h^* \Theta_h^*) \to 0$*.*

Proof. The f-divergence between Φ and Φ^* is as follows:

$$D_f(\Phi||\Phi^*) = \sum_{c_1=1}^{n_v^{(1)}} \cdots \sum_{c_{n_a}=1}^{n_v^{(n_a)}} \Phi_{c_1 \cdots c_{n_a}} f\left(\frac{\sum_{h=1}^{k} \pi_h^* \Theta^*_{h,c_1 \cdots c_{n_a}}}{\Phi_{c_1 \cdots c_{n_a}}}\right).$$

According to Lemma 5.1, Φ^* is the L_1 neighbor around Φ^0 with a probability of 1 when the amount of data approaches infinite, which implies that

$$\lim_{n_o \to \infty} \frac{\sum_{h=1}^{k} \pi_h^* \Theta^*_{h,c_1 c_2 \cdots c_{n_a}}}{\Phi_{c_1 c_2 \cdots c_{n_a}}} \to 1,$$

with a probability of 1. In addition, $f(1) = 0$ per the definition of f-divergence. Therefore,

$$\lim_{n_o \to \infty} f\left(\frac{\sum_{h=1}^{k} \pi_h^* \Theta^*_{h,c_1 \cdots c_{n_a}}}{\Phi_{c_1 \cdots c_{n_a}}}\right) \to 0.$$

As a result,

$$\lim_{n_o \to \infty} D(\Phi||\sum_{h=1}^{k} \pi_h^* \Theta_h^*) \to 0.$$

■

Theorem 5.2. *Given a non-IID categorical data set* $\mathbf{X} = \{\mathbf{x}_i \in \Pi^{n_a} | 1 \le i \le n_o\}$ *drawn from* Φ*, let the non-BEND-embedded data set be* $\mathbf{X}^* = \{\mathbf{x}_i^* \in \mathbb{R}^{k n_a} | 1 \le i \le n_o\}$*; there exists a function* $G(\cdot) : \mathbb{R}^{n_o \times k n_a} \to \Pi$ *so that, for any f-divergence* $D_f(\cdot||\cdot)$*,* $\lim_{n_0 \to \infty} D_f(\Phi||G(\mathbf{X}^*)) \to 0$*.*

According to Lemma 5.1, Lemma 5.2 can be obtained. Theorem 5.2 can then be proved based on Lemma 5.2.

Proof. Let $\mathbf{Y}^*$ be the unique elements in $\mathbf{X}^*$ and $G(\mathbf{X}^*) = \sum_{h=1}^{k} \mathbf{Y}_h^{*(1)} \circ \cdots \circ \mathbf{Y}_h^{*(n_a)}$, where $\circ$ refers to the Hadamard product and $\mathbf{Y}_h^{*(i)}$ is the attribute vector in $\mathbf{Y}$ according to the i-th attribute of the original data in h-th subspace. Since non-BEND models the generation process of non-IID categorical data and uses Equation (5.32) to represent the embedded data, $\mathbf{Y}_h^{*(1)} \circ \cdots \circ \mathbf{Y}_h^{*(n_a)} = \pi_h^* \Theta_h^*$ when the number of data objects approaches infinite. That is, $G(\mathbf{X}^*) = \sum_{h=1}^{k} \pi_h^* \Theta_h^*$. Hence, $\lim_{n_o \to \infty} D_f(\Phi || G(\mathbf{X}^*)) = \lim_{n_o \to \infty} D_f(\Phi || \sum_{h=1}^{k} \pi_h^* \Theta_h^*) \to 0$, according to Lemma 5.2, where $D_f(\cdot||\cdot)$ can be any f-divergence function. ■

Theorem 5.2 shows that the representation of categorical data by non-BEND is reversible with a large amount of data. Specifically, when the data size is infinite, the distribution reversed from the embedded data equals the original categorical data distribution under any f-divergence (e.g., the Kullback–Leibler divergence, the exponential divergence, and Kagan's divergence). With this property, the embedded data inherits information from the non-IID categorical data with bounded loss, and the resultant representation has good generalizability for different learning tasks.

5.4.2 Computational Complexity of Non-BEND

For the Gibbs sampling-based implementation, the most time-consuming part of non-BEND is Step 3 in Algorithm 1, which updates the conditional probability of subspace assignment according to Equation (5.11). Specifically, in each epoch of sampling, the algorithm counts the occurrence of each attribute value in every attribute for all objects in every IID subspace. Hence, the runtime cost of this step is linearly related to the number of values in each attribute, the number of attributes, the number of objects, and the number of latent IID subspaces. Because Step 3 repeats until the Gibbs sampling converges, the convergence speed of Gibbs sampling also affects the efficiency of non-BEND. Formally, the time complexity of non-BEND is $O(n_o n_a n_{mv} k n_{epoch})$, where n_{epoch} refers to the number of epochs until the Gibbs sampling converges. Since the number of clusters exhibits approximately logarithmic growth with the increase of the number of objects by the Dirichlet process, the time complexity of the proposed non-BEND algorithm is around $O(n_o n_a n_{mv} \log(n_o) n_{epoch})$. Actually, the number of IID subspaces is much smaller than that of objects and is thus ignorable, and the collapsed Gibbs sampling theoretically enjoys a fast convergence speed (Liu 1994).

For the VI-based implementation, the most time-consuming part of non-BEND is Steps 2–12 in Algorithm 2. Therefore, the time complexity of non-BEND is $O(n_o n_a n_t n_i)$, where n_t refers to the truncation level and n_i refers to the maximum iteration until convergence.

5.5 Connections between Non-BEND and the Existing Representation Methods

Non-BEND has strong connections to existing representation and similarity learning methods. When the number of attributes equals 1, non-BEND is simplified as the latent Dirichlet allocation (LDA) model (Blei et al. 2003). In this case, each subspace corresponds to a topic in LDA, and the distribution of each subspace is the word frequency in each topic. When attributes are independent of each other, the information captured by non-BEND is similar to the intra-attribute couplings in the COS (Wang, Dong, Zhou, Cao & Chi 2015) and other frequency-based methods (e.g., OF; Boriah et al., 2008). Both non-BEND and the context-based distance embedding methods (e.g., inter-attribute couplings in COS and the inter-attribute conditional probability in DILCA; Ienco et al., 2012) embed categorical data according to subspace distributions. The latter captures attribute interactions per the probability of an attribute value conditional on the values of other attributes. It is equivalent to embedding data per distributions in subspaces with respect to every attribute's values. These subspaces are more or less dependent on each other due to the relations between values. However, non-BEND uses distributions in independent subspaces to embed the original data, which reduces information duplication and disentangles the coupled factors. As a result, non-BEND disentangles the complex couplings and heterogeneities in non-IID data and enables the data representation to be automatically determined and adjusted for categorical data.

5.6 Experiments and Evaluation of Non-BEND Performance

Non-BEND is compared with (1) a baseline categorical data representation method one-hot embedding ("one-hot", for short) and its equivalent Hamming distance ("Hamming", for short); (2) five state-of-the-art categorical data representation methods including

CDE (Jian et al. 2017), COS (Wang, Dong, Zhou, Cao & Chi 2015), DILCA (Ienco et al. 2012), Ahmda (Ahmad & Dey 2007), and Rough (Cao, Liang, Li, Bai & Dang 2012); and (3) a deep representation method AE. For the AE, the first step is to transform categorical data to one-hot representation and then use three-layer fully connected network with 10 hidden nodes in each layer to implement. For all other methods, the recommended parameter settings are adopted.

5.6.1 Testing Non-BEND Effectiveness

5.6.1.1 Non-BEND-Enabled Clustering Performance

The experiments evaluate non-BEND with different non-IID categorical data decomposing methods, including collapsed Gibbs sampling (NB-G) and VI (NB-V). The experiments feed the representations learned by vector-based representation methods, including non-BEND, AE, and CDE, into k-means, which is probably the most popular clustering method and is sensitive to distance measure. Meanwhile, the experiments incorporate the representations learned by similarity-based representation methods, including COS, DILCA, Ahmad, Rough, and Hamming, into k-modes, which is the most commonly used clustering method for categorical data. To eliminate the randomness caused by random initial centers, k-means and k-modes are run 100 times on each data set and report the F-score of k-means and k-modes with the lowest objective value.

The clustering performance enabled by different categorical data representation methods are shown in Table 5.1. The best results are boldfaced. To avoid the impact of class imbalance, the experiments evaluate the performance by F-score (%), which is a combination of recall and precision. The higher F-score indicates better learning performance. It shows non-BEND (either NB-G and NB-V) performs significantly better than the compared methods on half of the data sets. On another half of the data sets, non-BEND also achieves the same or comparable performance as other methods. Because all data representation methods in the experiments are not task specific, none of them can always beat the others in all data sets when their ground-truth labels are for specific tasks. However, the overall performance of their enabled clustering reflects the capability of the representation method in revealing the intrinsic data characteristics. It is because a better representation should capture more data characteristics, which will at least be partially reflected by the ground truth.

To statistically compare non-BEND's performance with the compared categorical data representation methods, their ARs are tested by the Friedman test and Bonferroni–

Table 5.1: F-score of K-means and K-modes Enabled by Different Categorical Data Representation Methods

Data	NB-G	NB-V	AE	CDE	COS	Ahmad	DILCA	Rough	Hamming
SoyS	**100.0**	**100.0**	87.04	**100.0**	**100.0**	**100.0**	**100.0**	98.05	98.05
Zoo	**80.65**	79.69	51.92	75.04	72.1	71.34	71.34	62.79	73.27
DNAP	**91.51**	89.58	58.91	61.61	49.24	49.92	85.85	63.2	52.68
Hay	35.17	**54.17**	43.31	52.85	38.98	33.76	32.87	38.92	33.06
Hep	69.24	**69.82**	66.04	**69.82**	46.29	66.72	65.13	59.21	59.21
Aud	**38.8**	38.02	24.51	32.18	27.71	35.38	31.77	22.36	29.05
Hsv	89.22	89.65	**90.95**	89.65	88.36	88.36	88.79	87.04	86.64
Spc	56.69	39.46	37.58	52.55	36.26	34.93	34.76	**57.63**	35.94
Mof	**59.05**	**59.05**	51.8	56.65	50.18	50.22	48.68	50.62	50.98
SoyL	**72.39**	70.61	61.16	62.19	60.1	56.84	59.42	46.41	55.31
Prim	**27.27**	25.84	23.39	23.43	19.81	23.65	21.76	22.38	26.19
Tr	73.01	**78.1**	50.47	54.63	35.32	35.32	35.32	65.19	54.22
Wcs	94.52	94.52	95.59	**96.2**	94.28	95.12	95.49	94.44	89.98
Crx	52.65	52.49	52.65	52.65	36.99	52.65	**79.29**	63.47	**79.29**
Br	94.16	94.16	**95.95**	95.2	93.56	94.89	95.25	94.37	93.27
Ma	82.04	80.39	81.32	81.66	80.06	81.66	**82.65**	80.67	81.5
Dmg	96.63	**96.95**	66.99	73.1	74.58	72.87	72.61	57.99	66.6
Tic	**62.61**	52.22	50.77	54.8	51.88	50.87	52.97	50.19	53.59
DNAN	**80.47**	77.91	75.93	51.14	41.91	46.68	59.18	43.28	41.44
Spc	85.75	85.75	83.06	**87.12**	31.31	47.34	45.87	42.79	42.48
Krv	51.19	51.19	51.1	51.03	46.72	**55.17**	**55.17**	53.73	53.86
Ld	66.48	**67.31**	36.62	48.03	53.91	51.83	61.08	32.65	28.82
Ms	82.91	82.91	**84.78**	82.83	82.91	82.86	82.39	78.18	82.29
Cnt	32.12	32.12	30.98	31.91	27.23	32.88	**33.14**	30.34	31.43
AR	**2.79**	3.23	5.19	3.73	6.88	5.25	4.29	6.58	6.44

Dunn test (Demšar 2006). The χ_F^2 of Friedman test is 58.40 associated with a p-value $9.56e^{-10}$. This result indicates that the performances of the compared methods are not equal. Further, the Bonferroni–Dunn test evaluates the critical difference (CD) between non-BEND and other methods, and it shows the CD at p-value < 0.1 is 1.97. As shown in Table 5.3, non-BEND achieves overall ARs 2.79 and 3.23 with the NB-G and NB-V implementations, which are much better than other representation methods. For example, the AR of NB-G is 0.94 better than that of the best state-of-the-art method CDE (3.73), and 3.65 better than Hamming (6.44). With regards to the CD, non-BEND's performance is significantly better than all state-of-the-art methods except CDE.

Non-BEND enables superior clustering performance because it disentangles the complex non-IID space and embeds the fundamental information in the decoupled I-

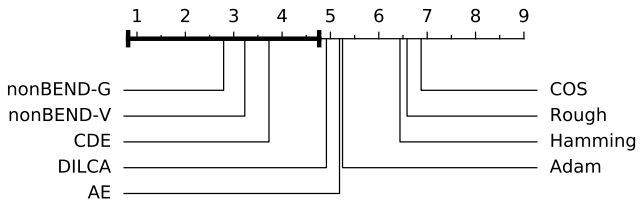


Figure 5.3: Comparison of clustering performance enabled by non-BEND against the other representation methods per the Bonferroni–Dunn test. All representation methods with ARs outside the marked interval are significantly different ($p < 0.1$) from non-BEND.

ID subspaces into its categorical data representation with limited information loss. For example, data sets DNAP and Tic have very high non-IID complexity, as reflected by the number of their NB-G-generated IID subspaces, which are 116 and 51, respectively. On data set DNAP, Non-BEND-enabled clustering achieves a 91.51% F-score, but the highest performance of its competitors is only 85.85%, and that of all others is under 63.2%. On Tic, non-BEND-enabled clustering achieves a 62.61% F-score, while the performance of its competitors is under 54.80%. With the embedded non-IID characteristics, non-BEND-represented data can enable much better clustering performance, especially on the data sets with high non-IID complexity. From this result, it can be also seen that the performance of non-BEND with VI is slightly worse than that of non-BEND with collapsed Gibbs sampling. This lower performance is because VI adopts simple distribution to approximate the original complex distribution, which causes some loss. However, considering VI is much faster than collapsed Gibbs sampling, the minor performance loss is acceptable.

5.6.2 Testing Non-BEND-Enabled Classification Performance

Further study is to evaluate the non-BEND representation performance in classification. In this experiment, it selects k-nearest neighbors (KNN) as the classifier, because KNN is a widely used classifier that heavily depends on data representation. Considering the representation effectiveness in terms of global distribution has been evaluated by the k-means or k-modes algorithm, this experiment aims to evaluate the local distribution of data representation by KNN. To achieve that evaluation, the class of an object should be determined only by a very small number of its nearest neighbors. Accordingly, this experiment uses cross-validation to select hyper-parameter k of KNN from $\{1,3,5,7\}$. For each data set, it randomly selects 90% of objects for training and uses

Table 5.2: F-score of KNN Enabled by Distance and Similarity Learning Methods

Data	NB-G	NB-V	AE	CDE	COS	Ahmad	DILCA	Rough	Hamming
SoyS	**100.0**	**100.0**	83.33	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Zoo	**100.0**	**100.0**	74.24	88.0	**100.0**	**100.0**	**100.0**	91.25	**100.0**
DNAP	**90.9**	89.23	75.24	80.48	73.06	77.14	87.78	74.94	80.9
Hay	74.56	67.45	45.89	71.09	80.76	57.69	57.58	**81.42**	51.64
Hep	67.94	69.45	**75.84**	65.24	64.05	65.4	61.73	64.01	66.55
Aud	48.35	44.52	19.98	42.03	45.8	51.56	**64.36**	40.17	40.59
Hsv	93.63	86.91	93.09	92.56	94.28	**95.81**	94.9	91.95	91.25
Spc	62.72	62.72	**68.17**	54.11	50.14	50.5	50.53	50.4	67.75
Mof	76.73	76.73	65.54	75.94	**82.94**	**82.94**	74.56	80.38	74.3
SoyL	93.29	91.84	83.02	93.16	**93.45**	93.43	92.87	90.05	91.28
Prim	**34.81**	34.49	27.19	33.87	23.09	28.3	27.46	20.8	31.7
Dmg	96.51	94.45	89.41	95.31	96.25	97.65	**98.08**	92.76	93.3
Tr	84.52	84.63	74.4	**90.86**	32.0	32.0	32.0	78.84	85.03
Wcs	96.06	96.38	**97.07**	96.49	93.9	95.99	94.61	92.77	96.12
Crx	84.75	82.82	**86.49**	84.64	79.17	82.79	81.37	77.69	82.41
Br	95.91	96.07	95.14	96.07	94.07	**96.34**	94.14	92.65	95.8
Ma	81.88	81.42	80.35	**82.52**	70.84	70.2	70.39	70.24	81.76
Tic	99.35	77.87	62.58	84.92	90.56	**100.0**	89.72	46.65	55.93
DNAN	**92.37**	89.72	91.58	83.98	77.52	80.33	91.65	81.46	80.87
Spc	**93.87**	90.24	92.37	86.45	77.25	79.85	84.96	81.05	79.21
Krv	**96.37**	**96.37**	84.79	95.0	91.77	92.46	91.39	89.0	95.38
Ld	67.97	64.55	55.59	**68.94**	62.11	61.81	62.58	47.89	62.5
Ms	**100.0**	**100.0**	**100.0**	**100.0**	99.98	**100.0**	**100.0**	**100.0**	**100.0**
AR	**2.87**	3.98	5.93	4.15	5.93	4.65	5.11	7.04	5.33

the remainder for testing, and 20 sampling iterations generate 20 sets of training and testing data for the experiment. The average classification performance is reported in Table 5.2.

The KNN classification results differ slightly from the k-means and k-modes clustering results in that non-BEND does not achieve the best result on most of data sets but always performs comparably to the best method on all data sets. Overall, it achieves 2.87 and 3.98 ARs with collapsed Gibbs sampling and VI, respectively. This overall performance is better than that of any other competitor.

Also, non-BEND is statistically compared with its competitors with the Friedman test and Bonferroni–Dunn test (Demšar 2006). The χ_F^2 of Friedman test is 41.40 associated with p-value $1.75e^{-6}$. This result indicates that the performance of all the compared methods is not equal. The Bonferroni–Dunn test shows the CD at p-value < 0.1 is 2.02. As demonstrated in Figure 5.4, the overall performance of non-BEND is

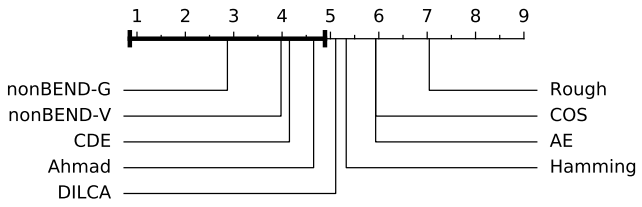


Figure 5.4: Comparison of classification performance enabled by non-BEND against the other representation methods per the Bonferroni–Dunn test. All representation methods with ARs outside the marked interval are significantly different ($p < 0.1$) from non-BEND.

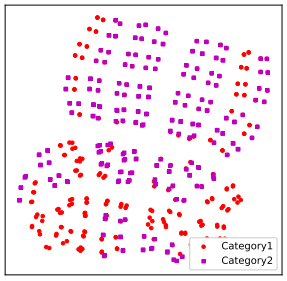
significantly better than that of other representation methods, except CDE and Ahmad.

The overall best classification performance led by non-BEND indicates that the non-BEND-represented data captures sufficient information from non-IID categorical data. However, non-BEND-enabled classification performance is slightly worse than non-BEND-enabled clustering performance. The main reason may be because non-BEND estimates the value distributions on decoupled IID subspaces and significantly embeds this information to provide a global description of categorical data. As a result, non-BEND-represented data may be better suited for a task that relies on global data distribution (e.g., k-means or k-modes clustering). Although the performance of non-BEND-enabled local distribution-based tasks (e.g., KNN classification) is also acceptable, we may further improve its performance by fine-tuning non-BEND representation for specific tasks. For example, training a neural network to transform non-BEND representation according to task labels. This improvement can be easily achieved because non-BEND representation embeds sufficient information in non-IID categorical data, with limited information loss.

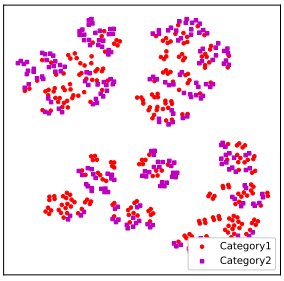
5.6.3 Testing Non-BEND Representation Quality

This section evaluates the non-BEND representation quality by comparing its visualization with other categorical data representation methods. To visualize the representations, this experiment coverts each of them from a high-dimensional vector to a two-dimensional vector by t-SNE (Maaten & Hinton 2008). The representations of different methods are visualized on data set Tr. The visualization is shown in Figure 5.5, in which different colors are used to represent different data categories.

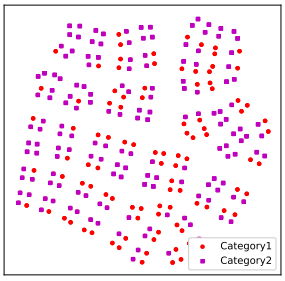
Compared with representations generated by competitors, non-BEND-represented



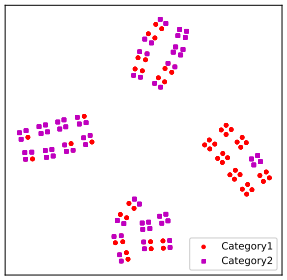
(a) Non-BEND represented distribution.



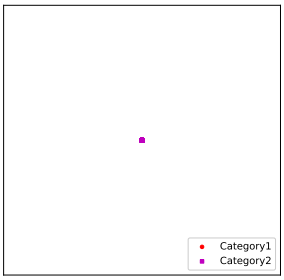
(b) AE represented distribution.



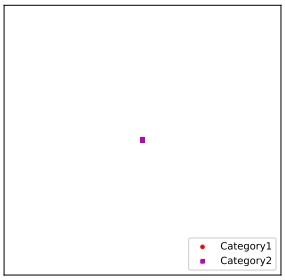
(c) CDE represented distribution.



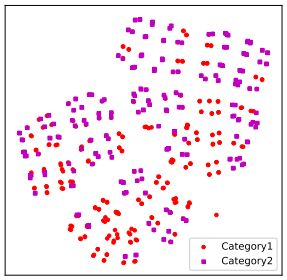
(d) COS represented distribution.



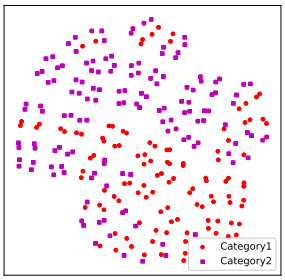
(e) Ahmad represented distribution.



(f) DILCA represented distribution.



(g) Rough represented distribution.



(h) One-hot represented distribution.

Figure 5.5: The visualization of different representation methods on data set Tr. These results illustrate the non-BEND-represented data shows clearer boundaries between different clusters. The plotted two-dimensional embedding is converted from high-dimensional representation by t-SNE. Different colors refer to different data categories per the ground truth.

data has a clearer structure where objects with the same category are located together. It qualitatively demonstrates that the non-BEND-represented data has good quality for learning tasks. It should be noticed that, in Figure 5.5, the representations of Ahmad and DILCA methods are collapsed to one point, which does not provide any useful information. This collapse is consistent with the clustering and classification results shown in Tables 5.1 and 5.2, where these two representations achieve only a 35.32% F-score and 32.00% F-score, respectively. The reason for this collapse is that both Ahmad and DILCA use only the frequency-based information to represent categorical data, but all values in Tr share the same frequency. In this case, Ahmad and DILCA fail to distinguish different categorical values. In contrast, non-BEND still works well on this kind of data, because it embeds the essential difference between categorical values by one-zero embedding in addition to frequency-based information.

5.6.4 Testing Non-BEND Flexibility

Non-BEND is flexible and can be fed into a variety of learning tasks that need categorical data representation. In addition to the above clustering and classification tasks, it further illustrates the flexibility and representation performance of non-BEND with a recommendation task. Following Zhao et al. (2017), it first applies non-BEND to represent the categorical attributes of users and items in a recommendation task. Then, based on the represented attributes, it adopts a Dirichlet Gaussian mixture model to cluster users and items. Lastly, a Bayesian matrix factorization (BMF) model (Salakhutdinov & Mnih 2008) recommends users and items be placed in the same clusters by sharing the same hyper-parameters. As a comparison, this experiment feeds the one-hot encoding to this model to evaluate whether non-BEND contributes to the recommendation performance. The experiment is conducted on a representative data set, MovieLens1M. The data set contains $1,000,209$ anonymous ratings (5-star scale) of $3,952$ movies by $6,040$ users. The user information contains gender, age range, occupation and zip code. The movie information includes the genres of each movie. All information is offered as categorical data.

This experiment sets the concentration parameter α of the Dirichlet process as 50 and the number of latent factors as 10. The hyper-parameters of Gaussian-Wishart are set as $\mu_0 = 0$, $\nu_0 = 10$, $\beta_0 = 2$, and W_0 as the identity matrix for both user and movie latent matrices. To gain insight into the experimental results, it further reports the number of user groups and item groups obtained based on the non-BEND-represented and one-hot encoding-represented information.

Table 5.3: The Accuracy of Bayesian Matrix Factorization Enabled by Different Representation Methods

		N-BMF	D-BMF	BMF
MovieLense1M	RMSE	**0.8413**	0.8423	0.8424
	MAE	**0.6589**	0.6597	0.6599
	#User Group	6	56	-
	#Item Group	11	42	-

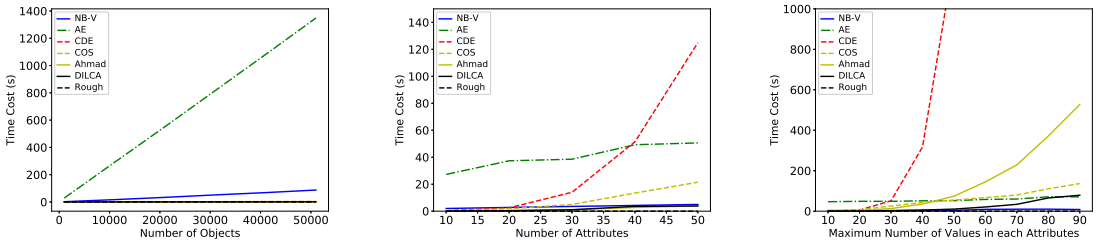
This experiment uses RMSE and MAE to evaluate the recommendation performance. Lower RMSE and MAE refer to more accurate recommendations. The experimental results are reported in Table 5.3. The non-BEND-enabled BMF (N-BMF) achieves the best performance, as compared with BMF and the one-hot encoding-enabled BMF (D-BMF). The results show that N-BMF improves performance in terms of both RMSE and MAE compared to BMF and D-BMF. This improvement further demonstrates the flexibility and effectiveness of non-BEND in different application scenarios.

5.6.5 Testing Non-BEND Efficiency

Here, synthetic data sets are generated to evaluate the efficiency of non-BEND in terms of the following data factors: the number of objects n_o, the number of attributes n_a, and the maximum number of values in each attribute n_{mv}. The default settings of these factors are as follows: n_o is $1{,}000$, n_a is 10, and n_{mv} is 3. The experiments generate three groups of data and tune one of these factors for each group. For the first group of data, the number of objects is adjusted from $1{,}000$ to $50{,}000$. For the second group of data, the number of attributes is tuned from 10 to 50. For the third group of data, the maximum number of values in attributes is changed from 10 to 90.

As shown in Section 5.4.2, the efficiency of non-BEND is mainly determined by three data factors: the number of objects (n_o), the number of attributes (n_a), and the maximum number of values in attributes (n_{mv}). The experiments evaluate the non-BEND efficiency per these three factors on the synthetic data sets. Since VI is much more efficient than is Gibbs sampling, this section reports only the time cost of NB-V. Furthermore, the time cost of other categorical data representation methods on the same data sets are reported, for comparison. The time cost of different categorical data representation methods under each data factor is shown in Figure 5.6.

In Figure 5.6, it is seen that non-BEND is linear to all the above data factors, including n_o, n_a, and n_{mv}. It demonstrates non-BEND is very efficient and has a good



(a) Time cost w.r.t. n_o. (b) Time cost w.r.t. n_a. (c) Time cost w.r.t. n_{mv}.

Figure 5.6: The non-BEND time cost with respect to data factors: object number n_o, attribute number n_a, and maximum number of attribute values n_{mv}.

scalability to large categorical data. This empirical result is consistent with the theoretical analysis of the non-BEND time complexity in Section 5.4.2.

The time cost of non-BEND and state-of-the-art categorical data representation methods is all linear to the number of objects n_o as shown in Figure 5.6(a). However, non-BEND is much more efficient than is the deep representation learning method AE. This increased efficiency is due to the many more parameters that need to be learned in the deep model, significantly increasing time cost. The time cost of non-BEND is much less than that of CDE in terms of n_a and n_{mv}, under which CDE has a quadratic time complexity while non-BEND has only linear time complexity, as shown in Figs. 5.6(b) and 5.6(c). Although the CDE-enabled clustering and classification performance does not significantly differ, as compared to that enabled by non-BEND, non-BEND can be used to represent categorical data with very high dimensionality and large amounts of unique categorical values, which CDE may not be able to represent due to very high time costs. Similar to CDE, Ahmad also shows comparable performance with non-BEND in the KNN classification task; however, it fails to represent categorical data with a large amount of unique categorical values because of its quadratic time complexity, as shown in Figure 5.6(c).

5.6.6 Ablation Study of Non-BEND Design

Further study is to test the contribution of estimating the characteristics of decoupled IID subspaces and the attribute weighting strategy. To achieve this goal, this experiment compares non-BEND with its variants without the attribute weighting strategy (NB-GU for Gibbs sampling implementation, and NB-VU for VI implementation) and one-zero embedding (OZ for short) with respect to their enabled k-means clustering per-

Table 5.4: F-score of K-means Enabled by Non-BEND and Its Variants

Data	NB-G	NB-GU	OZ	Δ_{GU}	Δ_{GO}	NB-V	NB-VU	OZ	Δ_{VU}	Δ_{VO}
SoyS	100.0	100.0	100.0	0.0	0.0	100.0	100.0	100.0	0.0	0.0
Zoo	80.65	78.79	76.52	1.85	4.13	79.69	77.49	76.52	2.2	3.17
DNAP	91.51	88.58	78.25	2.93	13.26	89.58	83.93	78.25	5.65	11.32
Hay	35.17	35.18	32.87	-0.0	2.3	54.17	34.64	32.87	19.53	21.3
Hep	69.24	66.72	69.24	2.52	0.0	69.82	69.82	69.24	0.0	0.58
Aud	38.8	37.37	31.77	1.43	7.03	38.02	35.11	31.77	2.9	6.25
Hsv	89.22	88.79	89.65	0.43	-0.43	89.65	89.22	89.65	0.43	0.0
Spc	56.69	54.72	52.98	1.97	3.71	39.46	54.19	52.98	-14.73	-13.52
Mof	59.05	56.65	56.65	2.39	2.39	59.05	56.65	56.65	2.39	2.39
SoyL	72.39	73.82	68.27	-1.43	4.12	70.61	72.44	68.27	-1.83	2.34
Prim	27.27	25.47	20.53	1.8	6.74	25.84	25.29	20.53	0.55	5.31
Tr	73.01	72.23	60.89	0.78	12.12	78.1	74.97	60.89	3.13	17.21
Wcs	94.52	94.52	94.22	0.0	0.3	94.52	94.52	94.22	0.0	0.3
Crx	52.65	52.65	52.65	0.0	0.0	52.49	52.65	52.65	-0.16	-0.16
Br	94.16	94.16	93.87	0.0	0.3	94.16	93.87	93.87	0.3	0.3
Ma	82.04	82.04	82.65	0.0	-0.61	80.39	82.65	82.65	-2.26	-2.26
Dmg	96.63	96.17	96.62	0.46	0.01	96.95	96.76	96.62	0.2	0.33
Tic	62.61	62.61	54.8	0.0	7.81	52.22	54.8	54.8	-2.58	-2.58
DNAN	80.47	78.02	75.55	2.45	4.92	77.91	79.16	75.55	-1.25	2.36
Spc	85.75	85.82	81.85	-0.07	3.9	85.75	85.54	81.85	0.21	3.9
Krv	51.19	51.19	51.19	0.0	0.0	51.19	51.19	51.19	0.0	0.0
Ld	66.48	68.73	59.18	-2.25	7.3	67.31	61.01	59.18	6.3	8.14
Ms	82.91	82.91	82.91	0.0	0.0	82.91	82.91	82.91	0.0	0.0
AR	**1.54**	1.93	2.52	0.39	0.98	**1.63**	1.85	2.52	0.22	0.89

formance. The clustering performance is reported in terms of F-score (%) in Table 5.4. The performance increase from NB-GU to NB-G (Δ_{GU}) and from NB-VU to NB-V (Δ_{VU}) reflects the contribution of the attribute weighting strategy. The performance increase from OZ to NB-GU (Δ_{GO}) and from OZ to NB-VU (Δ_{VO}) reflects the contribution of characteristics estimation on the decoupled subspaces.

As shown in Table 5.4, the non-BEND model achieves the best overall performance compared with its variants, which demonstrates the effectiveness of the non-BEND design. Regarding the contribution of the characteristics estimation in decoupled IID subspaces, the overall performance increase Δ_{GO} is 0.98 and Δ_{VO} is 0.89 in terms of the AR. These are very large improvement, since the AR ranges from 1 to 3. Regarding the contribution of the attribute weighting strategy, the overall performance increase Δ_{GU} is 0.39 and Δ_{VU} is 0.22 in terms of the AR. Although the weighting strategy does not enable the best performance on all data sets, it still contributes to the overall represen-

tation performance.

5.7 Summary

Complex categorical data presents intricate couplings and heterogeneities within and between attributes and objects. This chapter primarily studies heterogeneity learning, especially heterogeneous distribution learning, for complex categorical data representation. Specifically, it proposes an effective decoupled non-IID learning framework, DNL, that represents non-IID categorical data in two steps: (1) capturing attribute couplings and heterogeneity by decomposing non-IID categorical data into IID subspaces, in which attributes are independent of each other and values within an attribute are identically distributed; and (2) learning the data characteristics in each IID subspace and interactions between IID subspaces to represent non-IID categorical data in a Euclidean space.

The DNL framework for non-IID categorical data representation is general and can be customized into diverse methods for specific tasks. On the one hand, the key steps in the procedure can be flexibly implemented in terms of different approaches; on the other hand, DNL can be customized according to specific strategies (e.g., non-negative factorization), to enable better performance for targeted learning tasks. This thesis instantiates the DNL framework by a nonparametric Bayesian embedding model for categorical data representation, built on a Dirichlet multivariate multinomial mixture model. It further theoretically analyzes the information loss of the non-IID representation. According to the theoretical result, the proposed method approaches zero representation information loss as the number of data increases. Furthermore, the proposed method is efficient to represent categorical data with a large number of attributes. Therefore, it shows significant improvement on high-dimensional complex non-IID categorical data representation in terms of both effectiveness and efficiency, as compared with state-of-the-art methods.

The heterogeneity learning studied in this chapter, together with the coupling learning presented in Chapter 4, builds the foundation of non-IID learning. Although coupling learning and heterogeneity learning both show significant performance in categorical data representation, they do not jointly consider the heterogeneous distributions and heterogeneous dependencies (i.e., non-IID-completeness). For categorical data with non-IID-completeness, more powerful non-IID representation learning methods are required to embed heterogeneous couplings while considering heterogeneities. Accordingly, this thesis will study non-IID-completeness learning methods for categorical data

representation in the next two chapters.

Part III

Non-IID-Completeness Learning

CHAPTER 6

HETEROGENEOUS METRIC LEARNING OF CATEGORICAL DATA WITH HIERARCHICAL COUPLINGS

6.1 Introduction

Complex categorical data has complicated couplings and heterogeneities between values, attributes, and objects (Cao 2015). However, most existing categorical data representation methods capture either couplings or heteroegneities (see details in Section 2.5), and thus, they cannot embed complete non-IID characteristics in their-generated representation. As a result of a lack of sufficient representation of couplings and heterogeneities, the learned representation is ineffective for approximating data complexities and measuring distances in a categorical space.

To tackle the above problem, this chapter studies non-IID-completeness learning based on the foundations of coupling learning (see details in Chapter 4) and that of heterogeneity learning (see details in 5). Specifically, this chapter introduces a novel data-driven heterogeneous metric learning with hierarchical couplings (HELIC) for representing categorical data. First, HELIC captures both low-level value-to-attribute and high-level attribute-to-class couplings to comprehensively reveal the hierarchical characteristics in categorical data. It captures the following interactions: (1) the intra-attribute couplings to measure the within-attribute similarities (such couplings reflect the value interactions within an attribute); (2) the inter-attribute couplings to measure

the between-attribute similarities (these couplings describe the interactions between attribute values conditional on other attributes); and (3) the attribute-class couplings to measure the attribute-class similarities (these couplings reveal the value distribution with respect to each class). Second, HELIC reveals heterogeneities across various types of couplings to identify their different local structures and distributions. Lastly, HELIC learns a heterogeneous metric based on the captured couplings and heterogeneity.

The key contributions of this chapter include the following:

- *Learning hierarchical couplings:* Several learning functions are proposed to represent the hierarchical couplings in categorical data; that is, the value-to-class (including value-to-attribute and attribute-to-class) couplings within and between categorical attributes and between attributes and classes. These data-driven hierarchical value-to-class couplings complement and enhance metric performance.

- *Learning heterogeneities:* A learning model is proposed to learn and integrate heterogeneous local relationships in categorical data. The hierarchical value-to-class couplings are incorporated into different kernel functions, and a transformed matrix and combination coefficient are learned for each kernel to reveal the corresponding distributions of specific values and class labels. The learned heterogeneous information represents the different local views of categorical data relationships.

- *Theoretical analysis of effectiveness:* This analysis proves the effectiveness of HELIC in terms of improving metric learning accuracy from a theoretical point of view. The learning generalization error bound is also analyzed. The theoretical results explain why HELIC effectively discloses the complex relationships in categorical data.

This chapter compares HELIC with five state-of-the-art distance measures on 30 data sets with different data characteristics. The experimental results show that HELIC significantly improves the learning performance.

The rest of this chapter is organized as follows. Section 6.2 introduces the proposed method. Section 6.3 presents the theoretical analysis of the HELIC properties. Section 6.4 demonstrates the HELIC performance by comparing it with existing categorical distance measures from a variety of aspects. Lastly, Section 6.5 concludes this chapter.

6.2 The HELIC Design

This section introduces the working mechanisms of HELIC and its components.

6.2.1 Problem Statement of Metric Learning

The metric learning for categorical data aims to learn a distance metric $d(\cdot,\cdot):\mathscr{O}\times\mathscr{O}\rightarrow\mathscr{R}_0^+$ for all categorical objects in a data set O that satisfies the following properties:

1. $d(\mathrm{o}_i,\mathrm{o}_j)+d(\mathrm{o}_j,\mathrm{o}_k)\geq d(\mathrm{o}_i,\mathrm{o}_k)$,
2. $d(\mathrm{o}_i,\mathrm{o}_j)\geq 0$, and
3. $d(\mathrm{o}_i,\mathrm{o}_j)=d(\mathrm{o}_j,\mathrm{o}_i)$.

Denoting the representation of objects in the learned metric space as $\mathbf{x}$ that $d(\mathrm{o}_i,\mathrm{o}_j)=\mathbf{x}_i\odot\mathbf{x}_j$, where $\odot$ refers to any kind of operations between two vectors. The learned distance metric should also minimize the divergence between the data distribution in the categorical space $\mathfrak{O}$ and the data distribution in the metric space $\mathfrak{X}$. Although measuring the divergence directly is difficult, the divergence can be approximated by involving certain side information (like class label) for specific problems. Given an approximate divergence measure $\widetilde{Div}(\cdot||\cdot)$ based on side information, the objective function of metric learning for categorical data can be formalized as follows:

$$\begin{aligned}\underset{\mathbf{x}}{\text{minimize}}\quad & \widetilde{div}(\mathfrak{O}||\mathfrak{X})\\ \text{subject to}\quad & \mathrm{o}\sim\mathfrak{O}\\ & \mathbf{x}\sim\mathfrak{X}\\ & d(\mathrm{o}_i,\mathrm{o}_j)=\mathbf{x}_i\odot\mathbf{x}_j.\end{aligned}\tag{6.1}$$

In this chapter, the proposed HELIC learns a metric for categorical data. It satisfies all metric properties and simultaneously captures the hierarchical couplings and heterogeneity in categorical data to minimize the divergence between data distributions in the categorical space and metric space.

6.2.2 The HELIC Framework

Figure 6.1 illustrates the framework of HELIC, which has a three-layer hierarchical structure for coupling learning, heterogeneity learning, and metric learning. At the coupling learning stage, HELIC maps categorical data into three coupling spaces: intra-attribute couplings, inter-attribute couplings, and attribute-class couplings, to reveal

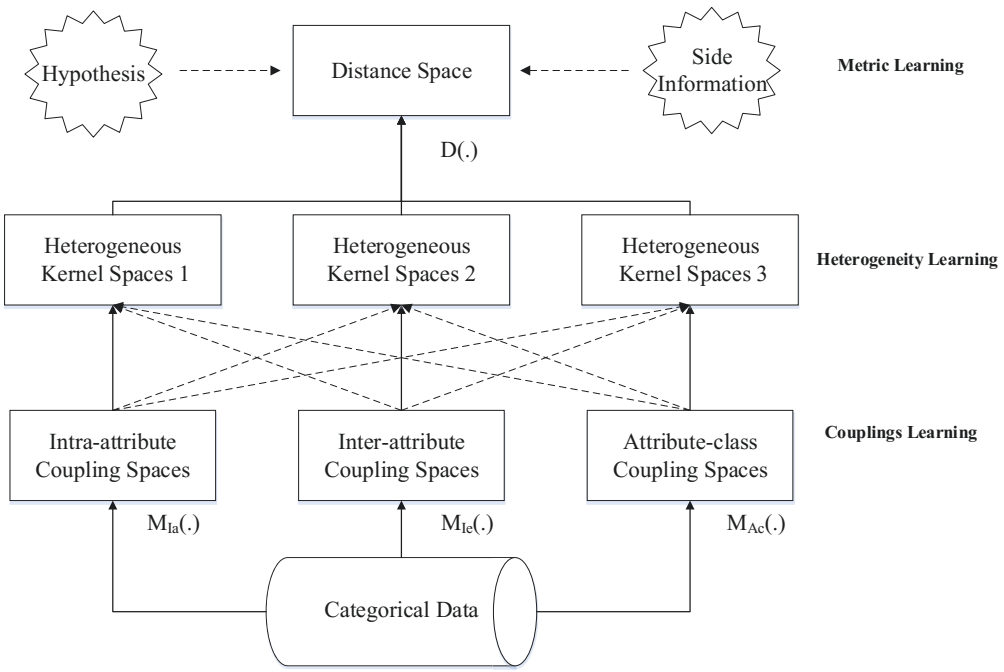


Figure 6.1: The HELIC framework: The coupling learning first represents the couplings in categorical data; then, the heterogeneity learning reveals the heterogeneous distributions of categorical values in the coupling spaces and feeds them into metric learning.

the value-to-class couplings. At the heterogeneity learning stage, HELIC models each type of coupling in the coupling spaces by specific kernel functions, and it learns a transformed matrix for each kernel to construct a heterogeneous kernel space. As a result, the various relationships corresponding to respective kernels are revealed. Lastly, at the metric learning stage, a distance metric is learned from the heterogeneous kernel spaces by involving a hypothesis or side information (e.g., label, ordinal, and semantic information) relevant to a learning task.

6.2.3 Learning Value-to-Class Couplings

Learning Intra-Attribute Couplings. *Intra-attribute couplings* represent the interactions between the values of an attribute. One way to observe such couplings is to analyze the value distributions in an attribute. This section sets a value frequency function to map the intra-attribute distributions to a numerical space, and the intra-attribute couplings are measured by the distance in this numerical space. For a categorical value

$\mathsf{v}_i^{(j)}$ in the j-th attribute, the intra-attribute coupling learning function $m_{Ia}^{(j)}(\mathsf{v}_i^{(j)})$ maps an intra-attribute coupling between the value and other categorical values in this attribute to a one-dimensional vector:

$$m_{Ia}^{(j)}(\mathsf{v}_i^{(j)}) = \frac{|g^{(j)}(\mathsf{v}_i^{(j)})|}{n_o}. \tag{6.2}$$

An intra-attribute coupling space is spanned by the vector obtained in an attribute by Equation (6.2) and is defined below:

$$\mathcal{M}_{Ia}^{(j)} = \{m_{Ia}^{(j)}(\mathsf{v}_i^{(j)}) | \mathsf{v}_i^{(j)} \in V^{(j)}\}. \tag{6.3}$$

For categorical data with n_a attributes, the intra-attribute coupling spaces $\mathcal{M}_{Ia}$ are as follows:

$$\mathcal{M}_{Ia} = \{\mathcal{M}_{Ia}^{(1)}, \cdots, \mathcal{M}_{Ia}^{(n_a)}\}. \tag{6.4}$$

An intra-attribute coupling space is a one-dimensional embedding of the categorical data space. Since a categorical value is often embedded in a high-dimensional attribute-value space, the intra-attribute coupling space cannot completely reflect the original attribute space. Inter-attribute couplings and attribute-class couplings are discussed in the following to capture the complementary information.

Learning Inter-Attribute Couplings. *Inter-attribute couplings* refer to the interactions between attributes, which contain the contextual or semantic information of values with respect to other attributes. For the examples in Table 2.1, if the numbers of $white$ and $black$ watermelons are similar, but the values of the other attributes (e.g., their root shapes) are differ significantly, these $white$ and $black$ colored watermelons can be differentiated by including their root shapes.

The HELIC method uses information conditional probability to present the inter-attribute couplings, revealing the categorical value distributions in subspaces with respect to values in other attributes. For a categorical value $\mathsf{v}_i^{(j)}$ in the j-th attribute and the set of values in other attributes $V_* = \{V^{(k)} | k \in N_a, k \neq j\}$, the inter-attribute coupling learning function is formalized as follows:

$$m_{Ie}^{(j)}(\mathsf{v}_i^{(j)}) = [p(\mathsf{v}_i^{(j)}|\mathsf{v}_{*1}), \cdots, p(\mathsf{v}_i^{(j)}|\mathsf{v}_{*k}), \cdots, p(\mathsf{v}_i^{(j)}|\mathsf{v}_{*|V_*|})]^\top, \tag{6.5}$$

where $\mathsf{v}_{*l} \in V_*$. With this learning function, the inter-attribute coupling space is constructed as follows:

$$\mathcal{M}_{Ie}^{(j)} = \{m_{Ie}^{(j)}(\mathsf{v}_i^{(j)}) | \mathsf{v}_i^{(j)} \in V^{(j)}\}. \tag{6.6}$$

For categorical data with n_a attributes, the inter-attribute coupling spaces $\mathcal{M}_{Ie}$ are as follows:

$$\mathcal{M}_{Ie} = \{\mathcal{M}_{Ie}^{(1)}, \cdots, \mathcal{M}_{Ie}^{(n_a)}\}. \tag{6.7}$$

The defined inter-attribute coupling mapping function reveals the value frequency in each subspace spanned by values in other attributes. The distance defined in the inter-attribute coupling space reflects inter-attribute couplings. The dimensionality of the inter-attribute coupling space equals $|V| - |V^{(j)}|$. The degree of freedom of the j-th attribute is $|V^{(j)}|-1$, which implies that the inter-attribute mapping function can project categorical values into a higher dimensional space if $|V| > 2|V^{(j)}| - 1$. In this case, the inter-attribute coupling space is powerful enough to describe the categorical attribute space.

Learning Attribute-Class Couplings. The *attribute-class couplings* capture the interactions between attributes and classes, which reveal the relationships between attribute value distributions with respect to each class. For a categorical value $\mathsf{v}_i^{(j)}$ in the j-th attribute, the learning function $m_{Ac}^{(j)}(\mathsf{v}_i^{(j)})$ adopts ICPF to reveal value distributions with respect to classes:

$$m_{Ac}^{(j)}(\mathsf{v}_i^{(j)}) = \left[\begin{array}{ccc} p(\mathsf{v}_i^{(j)}|c_1) & \cdots & p(\mathsf{v}_i^{(j)}|c_{n_c}) \end{array} \right]^\top. \tag{6.8}$$

The inter-attribute coupling space is an n_c-dimensional space that is spanned by the vector obtained by Equation (6.8) as follows:

$$\mathcal{M}_{Ac}^{(j)} = \{m_{Ac}^{(j)}(\mathsf{v})|\mathsf{v} \in V^{(j)}\}. \tag{6.9}$$

For categorical data with n_a attributes, the attribute-class coupling spaces are $\mathcal{M}_{Ac} = \{\mathcal{M}_{Ac}^{(1)}, \cdots, \mathcal{M}_{Ac}^{(n_a)}\}$.

Let us explain the need to consider attribute-class couplings. Assume the number of students who pass an exam is equal to those who fail the exam; then the similarity between passes and fails is very high when only the intra-attribute coupling is considered, indicating that the two outcomes, pass and fail, are similar. In reality, this similarity may not make sense; with the attribute-class couplings, a more reasonable understanding can be obtained. If students are classified as first or second class based on their overall performance, many more passes than fails appear in the first group, compared to fewer passes in the second group. This result shows that passes and fails are not highly coupled across all classes and that it is necessary to explore attribute-class couplings that can complement intra-attribute couplings.

6.2.4 Learning Heterogeneity in Heterogeneous Couplings

The learned couplings preserve the basic characteristics and different relationships of categorical values in the generated spaces. Although these couplings reveal low-level value relationships from various aspects, the high-level complex relationships among coupling spaces require in-depth analysis. Specifically, the heterogeneities in these coupling spaces should be learned, and the interactions among these coupling spaces should be disentangled.

The heterogeneities in coupling spaces refer to their various value distributions and different value relationships. Intrinsically, such heterogeneities are caused by two factors: (1) Each attribute may have a different value distribution, and values in an attribute may have different distributions; (2) each coupling may reflect a different kind of relationship. Learning the heterogeneous distributions can capture the difference between attributes and reveal the local structure in each attribute. Meanwhile, learning heterogeneous relationships can capture the couplings that are consistent with the final task while filtering the noise caused by the couplings that deviated from the final task.

The disentangled interactions between coupling spaces assist in discovering the independent components, and combining them can obtain the complete information in coupling spaces without redundancy. Value-to-class couplings capture hierarchical value relationships with regard to other values, attributes, and classes. In order to combine these relationships that are at different levels, a further transformation is needed to map them in the same space. Meanwhile, these couplings are not extremely independent. They may contain consensus and complementary information. To reduce the duplicated information, the coupling spaces should be selectively combined according to the interactions among them.

The HELIC method learns heterogeneity by adopting a variety of kernels to map heterogeneous coupling spaces into homogeneous kernel spaces and learning an adaptive kernel combination to integrate information with sparse regularization. The intuition behind this process is that different kernels are sensitive to different distributions. If the impact of a kernel on the values that match its sensitive local distribution can be preserved while the impact on others can be released, heterogeneous distribution learning can be wrapped into kernel learning. Meanwhile, the side information for the final learning task can be used to guide the kernel learning. Therefore, the relationships related to the final task can also be learned. As an effect of sparse regularization, the duplicated information in a kernel space can be reduced.

Given a kernel function $k(\cdot,\cdot)$ and a coupling space for the j-th attribute $\mathscr{M}^{(j)}$, denot-

ing the vector in the coupling space corresponding to value $\mathsf{v}_i^{(j)}$ as m_i, i.e., $\mathbf{m}_i = m^{(j)}(\mathsf{v}_i^{(j)})$, the kernel space is constructed by mapping each value pair as follows,

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{m}_1,\mathbf{m}_1) & k(\mathbf{m}_1,\mathbf{m}_2) & \cdots & k(\mathbf{m}_1,\mathbf{m}_{n_v^{(j)}}) \\ k(\mathbf{m}_2,\mathbf{m}_1) & k(\mathbf{m}_2,\mathbf{m}_2) & \cdots & k(\mathbf{m}_2,\mathbf{m}_{n_v^{(j)}}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{m}_{n_v^{(j)}},\mathbf{m}_1) & k(\mathbf{m}_{n_v^{(j)}},\mathbf{m}_2) & \cdots & k(\mathbf{m}_{n_v^{(j)}},\mathbf{m}_{n_v^{(j)}}) \end{bmatrix}.$$

Using various kernel functions for value-to-class coupling spaces, a set of kernel matrices $\{\mathbf{K}_1,\cdots,\mathbf{K}_{n_k}\}$ can be obtained. Further, a set of transformation matrices $\{\mathbf{T}_1,\cdots,\mathbf{T}_{n_k}\}$ can be learned to guarantee that the space of the p-th transformed kernel $\mathbf{K}'_p$ contains only the p-th kernel sensitive information, where $\mathbf{K}'_p$ is defined as

$$\mathbf{K}'_p = \mathbf{T}_p \cdot \mathbf{K}_p. \tag{6.10}$$

The HELIC method forces the transformation matrix $\mathbf{T}_p$ to be a diagonal matrix. In this case, the diagonal values of $\mathbf{T}_p$ are the weights of each categorical value for the p-th kernel. Let $\mathbf{T}_{p,ij}$ denote the value of the (i,j)-th entry of matrix $\mathbf{T}_p$. The large $\mathbf{T}_{p,ii}$ implies that the p-th kernel is more sensitive to the i-th value. Consequently, the spaces spanned by these transformed kernels are heterogeneous kernel spaces. To capture the relationships within the value-to-class coupling spaces and enhance the fitness of the distance measure in heterogeneous kernel spaces, HELIC wraps the above heterogeneity learning into metric learning to comprehensively learn the kernel transformation matrix and the distance measure. The metric learning method will be discussed in the next section.

6.2.5 Learning HELIC Representation

To learn a suitable distance measure for data in heterogeneous kernel spaces, HELIC uses the squared Euclidean distance in each heterogeneous kernel space as the base distance measure, and it then combines them to construct a suitable distance measure in heterogeneous kernel spaces.

Given a categorical data set, considering the p-th kernel matrix corresponding to the q-th attribute, let i and j represent the index of values in the p-th kernel space corresponding to the i-th and j-th objects, respectively. Specifically, $\mathsf{v}_\mathsf{i}^{(q)} = v^{(q)}(\mathsf{o}_i)$ and $\mathsf{v}_\mathsf{j}^{(q)} = v^{(q)}(\mathsf{o}_j)$. The distance between o_i and o_j in the p-th heterogeneous kernel space is $d_{p,ij}$:

$$d_{p,ij} = (\mathbf{K}'_{p,\mathsf{i}\cdot} - \mathbf{K}'_{p,\mathsf{j}\cdot})^\top (\mathbf{K}'_{p,\mathsf{i}\cdot} - \mathbf{K}'_{p,\mathsf{j}\cdot}), \tag{6.11}$$

where $\mathbf{K}'_{p,\mathrm{i}\cdot}$ and $\mathbf{K}'_{p,\mathrm{j}\cdot}$ are the i-th and j-th columns of $\mathbf{K}'_p$, respectively. As shown in Equation (6.11), the base distance $d_{p,ij}$ is determined by both the given kernel and the learned transformation matrix. In addition, it equals a squared Mahalanobis distance in its original kernel space,

$$d_{p,ij} = (\mathbf{K}_{p,\mathrm{i}\cdot} - \mathbf{K}_{p,\mathrm{j}\cdot})^\top \mathbf{T}_p^\top \mathbf{T}_p (\mathbf{K}_{p,\mathrm{i}\cdot} - \mathbf{K}_{p,\mathrm{j}\cdot}). \tag{6.12}$$

The distance metric d_{ij} between the i-th and j-th objects is defined by a linear combination of base distance measures from heterogeneous kernel spaces:

$$d_{ij} = \sum_{p=1}^{n_k} \alpha_p d_{p,ij}, \tag{6.13}$$

where α_p is the weight for the p-th base distance measure.

With a positive semi-definite matrix $\boldsymbol{\omega}_p = \alpha_p \mathbf{T}_p^\top \mathbf{T}_p$, the metric d_{ij} is calculated as follows:

$$d_{ij} = \sum_{p=1}^{n_k} \mathbf{k}_{p,ij}^\top \boldsymbol{\omega}_p \mathbf{k}_{p,ij}, \tag{6.14}$$

where $\mathbf{k}_{p,ij} = \mathbf{K}_{p,\mathrm{i}\cdot} - \mathbf{K}_{p,\mathrm{j}\cdot}$. Further, a vector is defined,

$$\mathbf{k}_{ij} = \begin{bmatrix} \mathbf{k}_{1,ij}^\top & \mathbf{k}_{2,ij}^\top & \cdots & \mathbf{k}_{n_k,ij}^\top \end{bmatrix}^\top, \tag{6.15}$$

and a diagonal matrix,

$$\boldsymbol{\omega} = \begin{bmatrix} \boldsymbol{\omega}_1^{\mathrm{diag}} & 0 & \cdots & 0 \\ 0 & \boldsymbol{\omega}_2^{\mathrm{diag}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{\omega}_{n_k}^{\mathrm{diag}} \end{bmatrix}, \tag{6.16}$$

where $\boldsymbol{\omega}_p^{\mathrm{diag}}$ is the diagonal matrix of $\boldsymbol{\omega}_p$. The metric d_{ij} is equal to a Mahalanobis distance in $\mathbf{k}_{ij}$'s space with a positive semi-definite matrix $\boldsymbol{\omega}$:

$$d_{ij} = \mathbf{k}_{ij}^\top \boldsymbol{\omega} \mathbf{k}_{ij}. \tag{6.17}$$

The learning of the set of kernel transformation matrices $\{\mathbf{T}_1, \cdots, \mathbf{T}_{n_k}\}$ and the combination coefficient of base distance measures $\{\alpha_1, \cdots, \alpha_{n_k}\}$ can be wrapped into the learning of a positive semi-definite matrix $\boldsymbol{\omega}$. In other words, to construct the metric space for categorical data, the only thing needed is to learn the positive semi-definite matrix

$\boldsymbol{\omega}$. Consequently, the data vector of the i-th object in the learned metric space can be represented by

$$\mathbf{x}_i = [\sqrt{\boldsymbol{\omega}_{1,11}}\mathbf{K}_{1,\mathrm{i}1}, \cdots, \sqrt{\boldsymbol{\omega}_{n_k,n_v^* n_v^*}}\mathbf{K}_{n_k,\mathrm{i}n_v^*}], \tag{6.18}$$

where $\boldsymbol{\omega}_{i,jj}$ is the value of the (j,j)-th entry in $\boldsymbol{\omega}_i$, and n_v^* is the number of values in the attribute corresponding to the n_k-th kernel. The learned representation $\mathbf{x}_i$ can be fed into a vector-based classifier as a numerical approximation of categorical data. Due to space limitations, this chapter uses only labels as the side information and assumes that the distance between objects with the same labels are smaller than the distance between objects with different labels. Matrix $\boldsymbol{\omega}$ should be sparse because each kernel is only sensitive to partial structure.

The learning objective function is defined as follows:

$$\begin{aligned} \underset{\boldsymbol{\omega},b}{\text{minimize}} \quad & \frac{1}{n_o^2}\sum_{i,j\in N_o}\xi_{ij} + \lambda\|\boldsymbol{\omega}\|_1 \\ \text{subject to} \quad & \boldsymbol{\omega} \succcurlyeq 0, \\ & \boldsymbol{\omega}_{kl} = 0 \quad for \quad k \neq l, \\ & 1 + r_{ij}(\mathbf{k}_{ij}^\top \boldsymbol{\omega}\mathbf{k}_{ij} - b) \leqslant \xi_{ij}, \\ & \xi_{ij} \geqslant 0, \forall i,j \in N_o. \end{aligned} \tag{6.19}$$

In the above function, $\|\cdot\|_1$ refers to the ℓ_1-norm, λ is a trade-off parameter that balances empirical error and regularization, and $\boldsymbol{\omega}_{kl}$ refers to the value of the (k,l)-th entry in the matrix $\boldsymbol{\omega}$. Value r_{ij} indicates whether two objects have the same label, defined as follows:

$$r_{ij} = \begin{cases} 1, & c(\mathrm{o}_i) = c(\mathrm{o}_j) \\ -1, & c(\mathrm{o}_i) \neq c(\mathrm{o}_j) \end{cases} \tag{6.20}$$

Since $\boldsymbol{\omega}$ is a diagonal matrix, this objective function can be efficiently optimized by linear programming.

This chapter uses the stochastic optimization method to obtain an approximate optimal solution for the learning objective function. As shown in Equation (6.19), the amount of training data is n_o^2 if there are n_o categorical objects. Although linear programming can be used to find the global optimal of Equation (6.19), it is time- and space-consuming when the amount of data is large. Instead of training all the data at once, HELIC uses the stochastic optimization method to randomly select a mini-batch of object pairs to calculate gradient values and update the learning parameters

at each iteration. After several iterations of training, HELIC reaches an approximate optimal that can construct the metric for categorical data. Section 6.3.3 will theoretically analyze the necessity and effectiveness of using stochastic optimization to obtain the approximate solution, and Section 6.4.5 will empirically evaluate the convergence of stochastic optimization for solving the objective function. Both theoretical and empirical analyses demonstrate that the stochastic optimization method can effectively solve the objective function with a fast convergence speed. Therefore, HELIC enjoys good scalability for large-scale categorical metric learning.

6.3 Theoretical Analysis of HELIC Properties

6.3.1 HELIC Effectiveness

The following lemma is given before proving the effectiveness of HELIC for categorical data.

Lemma 6.1. *Given an ideal kernel space spanned by a set of kernels $\mathscr{K} = \{\mathbf{K}_1, \cdots, \mathbf{K}_n\}$, the target similarity in categorical space $\mathscr{S}$, and a bounded loss function $\ell(\cdot)$, the expected loss of similarity with respect to $\mathscr{S}$ based on $\mathscr{K}$ and its subset $\mathscr{K}_*$ are denoted by $\hat{\ell}(\mathscr{S};\mathscr{K})$ and $\hat{\ell}(\mathscr{S};\mathscr{K}_*)$. Their difference is bounded as*

$$|\hat{\ell}(\mathscr{S};\mathscr{K}) - \hat{\ell}(\mathscr{S};\mathscr{K}_*)| \leq \sqrt{I(\mathscr{S};\mathscr{K} \setminus \mathscr{K}_* | \mathscr{K}_*)}, \tag{6.21}$$

where $I(\cdot;\cdot|\cdot)$ is the conditional mutual information, and $\mathscr{K} \setminus \mathscr{K}_*$ refers to the subspace that is spanned by the set of kernels in $\mathscr{K}$ minus the set of kernels in $\mathscr{K}_*$.

Proof. Considering $|\ell(x)| \leq 1$, and two probability distributions P and Q, thus,

$$\begin{aligned} \left| \int \ell(x) dQ - \int \ell(x) dP \right| &= \left| \int (1-\theta) \ell(x) dQ \right| \\ &\leq \int |1-\theta| \, dQ \\ &\leq \sqrt{KL(Q;P)}, \end{aligned} \tag{6.22}$$

where $\theta = \frac{dP}{dQ}$, $KL(\cdot;\cdot)$ is the KL divergence, and the upper bound holds because the ℓ_1 variational distance is bounded by the square root of the KL divergence.

With the above, for a fixed ideal similarity set $\mathscr{K}$, its subspace $\mathscr{K}_*$ and the target similarity space $\mathscr{S}$, it can be obtained that

$$|E_{\mathscr{S}|\mathscr{K}} \ell(\mathscr{S};\mathscr{K})| - E_{\mathscr{S}|\mathscr{K}_*} \ell(\mathscr{S};\mathscr{K}) \leq \sqrt{KL(P_{\mathscr{S}|\mathscr{K}}; P_{\mathscr{S}|\mathscr{K}_*})}, \tag{6.23}$$

where $P_{\mathscr{S}|\mathscr{K}}$ and $P_{\mathscr{S}|\mathscr{K}_*}$ are the conditional distribution of $\mathscr{S}$ conditioned on $\mathscr{K}$ and $\mathscr{K}_*$, respectively.

Taking the expectation with respect to $\mathscr{K}$ and Jensen's inequality, it can be obtained that,

$$|\hat{\ell}(\mathscr{S};\mathscr{K}) - E_{\mathscr{K}} E_{\mathscr{S}|\mathscr{K}_*} \ell(\mathscr{S};\mathscr{K})| \leq \sqrt{E_{\mathscr{K}} KL(P_{\mathscr{S}|\mathscr{K}}; P_{\mathscr{S}|\mathscr{K}_*})}. \tag{6.24}$$

Since

$$\hat{\ell}(\mathscr{S};\mathscr{K}_*) \leq E_{\mathscr{K}} E_{\mathscr{S}|\mathscr{K}_*} \ell(\mathscr{S};\mathscr{K}) \tag{6.25}$$

and

$$\hat{\ell}(\mathscr{S};\mathscr{K}) \leq \hat{\ell}(\mathscr{S};\mathscr{K}_*), \tag{6.26}$$

thus,

$$|\hat{\ell}(\mathscr{S};\mathscr{K}) - \hat{\ell}(\mathscr{S};\mathscr{K}_*)| \leq \sqrt{E_{\mathscr{K}} KL(P_{\mathscr{S}|\mathscr{K}}, P_{\mathscr{S}|\mathscr{K}_*})}. \tag{6.27}$$

Considering

$$E_{\mathscr{K}} KL(P_{\mathscr{S}|\mathscr{K}}, P_{\mathscr{S}|\mathscr{K}_*}) = I(\mathscr{S};\mathscr{K} \setminus \mathscr{K}_* | \mathscr{K}_*), \tag{6.28}$$

thus,

$$|\hat{\ell}(\mathscr{S};\mathscr{K}) - \hat{\ell}(\mathscr{S};\mathscr{K}_*)| \leq \sqrt{I(\mathscr{S};\mathscr{K} \setminus \mathscr{K}_* | \mathscr{K}_*)}. \tag{6.29}$$

■

Therefore, the HELIC effectiveness is ensured by the following theorem.

Theorem 6.1. *HELIC can improve the metric accuracy if each coupling space contains information complementary to the other coupling spaces and if the heterogeneous kernel spaces capture such information effectively.*

Proof. Given a set of coupling spaces

$$\mathscr{M} = \{\mathscr{M}_1, \mathscr{M}_2, \cdots, \mathscr{M}_{n^*}\},$$

and a set of kernel matrices $\{\mathbf{K}_1, \cdots, \mathbf{K}_n\}$ generated on the coupling spaces, let the kernel spaces spanned by these kernel matrices be $\mathscr{K}_* = \{\mathbf{K}_1, \cdots, \mathbf{K}_n\}$ and the target similarity

space be $\mathscr{S}$. The kernel space $\mathscr{K}_*$ must belong to an ideal space $\mathscr{K}$ that obeys the following property:

$$I(\mathscr{S};\mathscr{K}_m|\mathscr{K}_{m-1}) > 0, \tag{6.30}$$

where $\mathscr{K}_m$ and $\mathscr{K}_{m-1}$ refer to the subspace in $\mathscr{K}$ that are spanned by a set of m components and a subset with $m-1$ elements in that set, respectively.

In addition,

$$\begin{aligned}
&I(\mathscr{S};\mathscr{K}\setminus\mathscr{K}_{m-1}|\mathscr{K}_{m-1}) - I(\mathscr{S};\mathscr{K}\setminus\mathscr{K}_m|\mathscr{K}_m)\\
&= (H(\mathscr{S},\mathscr{K}_{m-1}) + H(\mathscr{K}) - H(\mathscr{S},\mathscr{K}) - H(\mathscr{S},\mathscr{K}_{m-1}))\\
&- (H(\mathscr{S},\mathscr{K}_m) + H(\mathscr{K}) - H(\mathscr{S},\mathscr{K}) - H(\mathscr{S},\mathscr{K}_m))\\
&= (H(\mathscr{K}_m) - H(\mathscr{K}_{m-1})) - (H(\mathscr{S},\mathscr{K}_m) - H(\mathscr{S},\mathscr{K}_{m-1}))\\
&= H(\mathscr{K}_m|\mathscr{K}_{m-1}) - H(\mathscr{K}_m|\mathscr{S},\mathscr{K}_{m-1})\\
&= I(\mathscr{S},\mathscr{K}_m|\mathscr{K}_{m-1}).
\end{aligned} \tag{6.31}$$

Therefore,

$$I(\mathscr{S};\mathscr{K}\setminus\mathscr{K}_{m-1}|\mathscr{K}_{m-1}) - I(\mathscr{S};\mathscr{K}\setminus\mathscr{K}_m|\mathscr{K}) > 0. \tag{6.32}$$

According to Lemma 6.1 and Equation (6.32), when m increases, $I(\mathscr{S};\mathscr{K}\setminus\mathscr{K}_m|\mathscr{K}_m)$ as well as the gap between $\ell(\mathscr{S};\mathscr{K})$ and $\ell(\mathscr{S};\mathscr{K}_m)$, will decrease. Therefore, the similarities in the kernel space $\mathscr{K}_*$ can increase the similarity accuracy. This means that the HELIC can improve the similarity accuracy if each coupling space contains information that is complementary to others. ■

Since HELIC adopts different coupling learning functions to capture data characteristics with respect to the intra-attribute value distributions, inter-attribute interactions, and attribute-class relationships, the information contained in each coupling space reflects a respective view and is thus complementary to other coupling spaces. The proposed heterogeneity learning approach reveals and combines different information from value-to-class coupling spaces. Hence, HELIC-based metric learning captures heterogeneous characteristics embedded in categorical data.

6.3.2 The Generalization Error Bound of HELIC

This section analyzes HELIC's generalization error bound, which shows to what extent HELIC can achieve its effectiveness. The HELIC's generalization error bound is given by the following theorem.

Theorem 6.2. *Let $\varepsilon(\boldsymbol{\omega}, b)$ be the expected error, and let $\varepsilon_{\mathcal{Z}}(\boldsymbol{\omega}, b)$ be the empirical error. For any $0 < \delta < 1$, the generalization error of learning can be bounded with probability $1-\delta$ as*

$$\begin{aligned}\varepsilon(\boldsymbol{\omega}, b) - \varepsilon_{\mathcal{Z}}(\boldsymbol{\omega}, b) \leqslant\, & 2(1+1/\sqrt{\lambda})\sqrt{2\ln(1/\delta)/n_o} \\ & + \left(8 + 16\sqrt{e\ln(n_o n_k)}\right)/\sqrt{n_o\lambda} + 12/\sqrt{n_o}.\end{aligned} \tag{6.33}$$

Proof. According to Theorem 3 in Cao et al. (2016), it can be obtained that

$$\begin{aligned}\varepsilon(\boldsymbol{\omega}, b) - \varepsilon_{\mathcal{Z}}(\boldsymbol{\omega}, b) \leqslant\, & \frac{4R_n}{\sqrt{\lambda}} + \frac{4(3 + 2X_*/\sqrt{\lambda})}{\sqrt{n_o}} \\ & + 2\left(1 + X_*/\sqrt{\lambda}\right)\left(\frac{2\ln\left(\frac{1}{\sigma}\right)}{n_o}\right)^{\frac{1}{2}},\end{aligned} \tag{6.34}$$

where R_n is the Rademacher complexity of metric learning defined as

$$R_n = \frac{1}{\lfloor \frac{n_o}{2} \rfloor} E_{\mathcal{K}} E_{\sigma} \left\| \sum_{i=1}^{\lfloor \frac{n_o}{2} \rfloor} \sigma_i \mathbf{k}_{i(\lfloor \frac{n_o}{2} \rfloor + 1)} \right\|_*,$$

and

$$X_* = \sup_{\mathbf{k} \in \mathcal{K}} \left\| \mathbf{k}_{ij} \mathbf{k}_{ij}^{\top} \right\|_*.$$

In the above formula, $\|\cdot\|_*$ refers to the dual norm of ℓ_1-norm, $\lfloor \frac{n_o}{2} \rfloor$ denotes the largest integer less than $\frac{n_o}{2}$, $\mathcal{K}$ is the kernel space, $\{\sigma_1, \sigma_2, \cdots, \sigma_{\lfloor \frac{n_o}{2} \rfloor}\}$ are independent variables drawn from the Rademacher distribution, and $E_{\mathcal{D}}$ is the expectation over space $\mathcal{D}$. Since the dual norm of ℓ_1-norm is the ℓ_∞-norm and the maximum of $\mathbf{k}_{ij}$ in the space $\mathcal{K}$ is not more than 1; hence,

$$X_* = \sup_{\mathbf{k} \in \mathcal{K}} \left\| \mathbf{k}_{ij} \right\|_{\infty}^2 \leqslant 1, \tag{6.35}$$

and the Rademacher complexity can be written as

$$R_n = \frac{1}{\lfloor \frac{n_o}{2} \rfloor} E_{\mathcal{K}} E_{\sigma} \left\| \sum_{i=1}^{\lfloor \frac{n_o}{2} \rfloor} \sigma_i \mathbf{k}_{i(\lfloor \frac{n_o}{2} \rfloor + i)} \right\|_{\infty}.$$

Considering the property of norm and letting $i^* = \left(\lfloor \frac{n_o}{2} \rfloor + i\right)$, for any $1 < q < \infty$, that

$$
\begin{aligned}
R_n &\leqslant \frac{1}{\lfloor \frac{n_o}{2} \rfloor} E_{\mathscr{K}} E_{\sigma} \left\| \sum_{i=1}^{\lfloor \frac{n_o}{2} \rfloor} \sigma_i \mathbf{k}_{ii^*} \right\|_q \\
&= \frac{1}{\lfloor \frac{n_o}{2} \rfloor} E_{\mathscr{K}} E_{\sigma} \left(\sum_{l=1}^{n_o n_k} \sum_{m=1}^{n_o n_k} \left| \sum_{i=1}^{\lfloor \frac{N_o}{2} \rfloor} \sigma_i \mathbf{k}_{l,ii^*} \mathbf{k}_{m,ii^*} \right|^q \right)^{\frac{1}{q}} \\
&\leqslant \frac{1}{\lfloor \frac{n_o}{2} \rfloor} E_{\mathscr{K}} \left(\sum_{l=1}^{n_o n_k} \sum_{m=1}^{n_o n_k} E_{\sigma} \left| \sum_{i=1}^{\lfloor \frac{n_o}{2} \rfloor} \sigma_i \mathbf{k}_{l,ii^*} \mathbf{k}_{m,ii^*} \right|^q \right)^{\frac{1}{q}},
\end{aligned} \tag{6.36}
$$

where $\mathbf{k}_{l,ij}$ refers to the l-th element of vector $\mathbf{k}_{ij}$ defined as in Equation (6.15). Considering the Khinchin–Kahane inequality (Kahane 1993),

$$
\begin{aligned}
& E_{\sigma} \left| \sum_{i=1}^{\lfloor \frac{n_o}{2} \rfloor} \sigma_i \mathbf{k}_{l,ii^*} \mathbf{k}_{m,ii^*} \right|^q \\
& \leqslant q^{\frac{q}{2}} \left(E_{\sigma} \left| \sum_{i=1}^{\lfloor \frac{n_o}{2} \rfloor} \sigma_i \mathbf{k}_{l,ii^*} \mathbf{k}_{m,ii^*} \right|^2 \right)^{\frac{q}{2}} \\
& = q^{\frac{q}{2}} \left(\sum_{i=1}^{\lfloor \frac{n_o}{2} \rfloor} \mathbf{K}^2_{l,ii^*} \mathbf{K}^2_{m,ii^*} \right)^{\frac{q}{2}} \\
& \leqslant \sup_{\mathbf{k} \in \mathscr{K}} \left\| \mathbf{k}_{ij} \right\|_{\infty}^{2q} \left(\left\lfloor \frac{n_o}{2} \right\rfloor \right)^{\frac{q}{2}} q^{\frac{q}{2}} \\
& \leqslant \left(\left\lfloor \frac{n_o}{2} \right\rfloor \right)^{\frac{q}{2}} q^{\frac{q}{2}}.
\end{aligned} \tag{6.37}
$$

Putting the result of Equation (6.37) into Equation (6.36), it can be obtained that

$$
R_n \leqslant \frac{1}{\lfloor \frac{n_o}{2} \rfloor} \left((n_o n_k)^2 \left(\left\lfloor \frac{n_o}{2} \right\rfloor \right)^{\frac{q}{2}} q^{\frac{q}{2}} \right)^{\frac{1}{q}}. \tag{6.38}
$$

Letting $q = 4\ln(n_o n_k)$, Equation (6.38) induces that

$$
\begin{aligned}
R_n &\leqslant 2\sqrt{e \ln(n_o n_k) / \left\lfloor \frac{n_o}{2} \right\rfloor} \\
&\leqslant 4\sqrt{e \ln(n_o n_k) / n_o}.
\end{aligned} \tag{6.39}
$$

Putting Equation (6.35) and Equation (6.39) into Equation (6.34) yields Equation (6.33). Theorem 6.2 is thus proved. ■

This bound provides a solid foundation for HELIC and offers the following guidance: (1) More training data leads to a lower generalization error. (2) The smaller the number of base kernels, the lower the generalization error. Fortunately, d has a logarithmic relationship with the generalization error. This relationship shows that the number of base kernels will not dramatically increase the generalization error. (3) According to Theorem 6.1, increasing the number of base kernels provides more complementary information which reduces the expected error $\varepsilon(\boldsymbol{\omega}, b)$. Therefore, the overall performance will still increase when complementary base kernels are added. This increase in performance can be achieved by building additional coupling spaces and using more discriminative kernels.

6.3.3 Computational Complexity of HELIC

The HELIC time complexity is determined by two parts: coupling learning and metric learning. For the coupling learning part, the time cost depends on what kind of couplings HELIC captures. In this chapter, HELIC captures the intra-attribute couplings, inter-attribute couplings, and attribute-class couplings. Calculating intra-attribute coupling requires the measurement of the frequency of each value, which has a complexity of $O(n_v)$. Capturing inter-attribute couplings requires calculating the relationship of each value pair in each attribute pair. Accordingly, the time cost is $O(n_{mv}^2 n_a^2)$, n_{mv} is the maximal number of values in the attributes. For the attribute-class couplings, HELIC needs to calculate the relation between each value and each class that has time complexity of $O(n_v n_c)$. Therefore, the time complexity for the coupling learning part is $O(n_v(n_c+1)+n_{mv}^2 n_a^2)$ in this chapter.

In metric learning, the time complexity depends on the solution. If using linear programming to find the global optima, the time complexity in the worst case scenario is $O((n_o^2)^{3.5} n_{\boldsymbol{\omega}}^2)$, and a fast approximate solution can achieve $O(n_o^2 + \frac{2(n_{\boldsymbol{\omega}}+n_o^2)\log(n_o^2)}{\epsilon^2})$ with $O(1\pm\epsilon)$ cost, where $n_{\boldsymbol{\omega}} = \sum_{i=1}^{n_k} n_v^{(i*)}$ is the length of parameter $\boldsymbol{\omega}$ and where $n_v^{(i^*)}$ refers to the number of values in attribute i^* corresponding to the i-th kernel. If using stochastic optimization for an approximate optimum, the time complexity is only $O(n_b n_{\boldsymbol{\omega}} n_{step})$, where n_b is the number of object pairs in each batch and where n_{step} is the number of iterations used to achieve convergence. Compared with the linear programming method, stochastic optimization is much more efficient if it can converge within a small number of iterations.

Actually, stochastic optimization can effectively find an approximate optimal solution for HELIC. Considering the structure of the objective function, the key parame-

ter which needs to be learned is $\boldsymbol{\omega}$, which is a vector with size $n_{\boldsymbol{\omega}}$. When $n_o^2 \gg n_{\boldsymbol{\omega}}$, the loss of the objective function will converge before using all n_o^2 training data. Since $n_{\boldsymbol{\omega}} = \sum_{i=1}^{n_k} n_v^{(i*)} = n_v n_k^*$, where n_k^* is a constant that relates to the number of used kernels, the above condition can be rewritten as $n_o^2 \gg n_v n_k^*$. Fortunately, the large scale categorical data always fits this condition because (1) the number of discrete values is much less than the number of objects and because (2) the number of base kernels used in HELIC should be small to guarantee a lower generalization error according to Section 6.3.2.

The space complexity of HELIC with linear programming is $O(n_o^2 n_{\boldsymbol{\omega}})$ and of HELIC with stochastic optimization is $O(n_b n_{\boldsymbol{\omega}})$. For a large categorical data set, the space complexity is very high when linear programming is used to find a perfect solution. However, a stochastic optimization-based approximate solution largely reduces the space complexity, since $n_b \ll n_o^2$. Therefore, it is necessary to use stochastic optimization for HELIC to tackle large categorical data.

Overall, HELIC has the time complexity $O(n_v(n_c+1) + n_{mv}^2 n_a^2 + n_b n_{\boldsymbol{\omega}} n_{step})$ and space complexity $O(n_b n_{\boldsymbol{\omega}})$. This means HELIC suits large data. In addition, HELIC can be further sped by applying parallel computing to both coupling learning and metric learning for large data.

6.4 Experiments and Evaluation of HELIC Performance

6.4.1 Parameter Settings of HELIC

In the experiments, the HELIC default settings are as follows. The kernels used in HELIC are 11 Gaussian kernels with width from 2^{-5} to 2^5 and three polynomial kernels with order from 1 to 3. The value for λ is set as $\frac{1}{n_k}$ for HELIC. The stochastic optimization method used to solve the HELIC objective function is Adam (Kingma & Ba 2014) with the initial learning rate 10^{-3}, the batch size of 20, and the number of iterations 1,000. For the parameters in these baseline methods, their recommended settings are taken here.

6.4.2 Testing HELIC Representation Performance

The HELIC representation performance is evaluated from the following perspectives: (1) The HELIC learned metric is compared with the baseline categorical distance measure (i.e., Hamming distance; "Hamming", for short). (2) It is also compared with five state-of-the-art distance measures for categorical data: COS (Wang, Dong, Zhou, Cao & Chi 2015), MTDLE (Zhang et al. 2015), Ahmad (Ahmad & Dey 2007), DILCA (Ienco et al. 2012), and Rough (Cao, Liang, Li, Bai & Dang 2012), to show whether HELIC outperforms the others in learning metric.

6.4.2.1 HELIC-Enabled Classification Performance

The distance learned by HELIC is incorporated into k-nearest neighbors (KNN), which is probably the most popular distance-based classifier and is sensitive to distance measures, to demonstrate the HELIC-enabled classification performance.

The HELIC method is compared with six distance measures with the results shown in Table 6.1. The averaged classification performance and standard deviation on the partitions of a data set are reported with respect to F-score. The best results are highlighted in bold; the results without significant difference from the best results under the *student t-test* (p-value > 0.05) are labelled by *, and Δ is the HELIC's improvement over the best results of other measures. The ARs of these methods over all data sets are reported to show their overall performance. It should be noted that MTDLE cannot produce the distance results on some large data sets in the experiments due to out-of-memory error, signaled by *NA* in the table. In most cases, HELIC performs significantly better than the compared methods. For example, the F-score improves 40.93% in Ba and 32.48% in Titn compared to the best-performing method MTDLE and COS. In some simple data sets (e.g., Zoo, Monk and Ms, on which the Hamming method achieves high performance), HELIC achieves 100.

The results also show that some of state-of-the-art measures fail to appropriately measure distances in some data sets. For example, on Monk and Ba, COS, Ahmad and DILCA achieve the same low performance. The reason lies in the fact that they capture only frequency-based intra- or inter-attribute couplings but ignore other interactions, for example, attribute-class couplings; values in these data sets follow a uniform distribution and have the same frequency. In contrast, HELIC captures hierarchical value-to-class couplings that address such data characteristics.

To statistically compare HELIC's performance with the above distance measures, it

Table 6.1: KNN Classification F-score (%) with Different Distance Measures. The Monte Carlo cross-validation results are reported with respect to *mean* ± *standard deviation*. The best results are highlighted in bold, the results without a significant difference from the best results for a data set under the *student t-test* (p-value > 0.05) are labelled by *, and Δ is the HELIC's improvement over the best results of the other measures. The AR of a method over all data sets with significant difference from others with respect to the *Bonferroni–Dunn test* (p-value < 0.05) is labelled by **.

Data	n_o	n_a	HELIC	COS	MTDLE	Ahmad	DILCA	Rough	Hamming	Δ
SoyS	47	35	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00** *	**100 ± 0.00***	**100 ± 0.00** *	**100 ± 0.00***	0.00%
Zoo	101	16	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00***	**100 ± 0.00***	97.75 ± 11.11	**100 ± 0.00***	0.00%
DNAP	106	57	**92.90 ± 5.85***	75.89 ± 13.35	81.67 ± 10.19	79.98 ± 9.14	90.33 ± 10.31	81.16 ± 10.30	78.05 ± 12.00	2.85%
Hay	132	4	**90.85 ± 5.07***	79.64 ± 9.71	68.54 ± 10.55	52.26 ± 10.20	54.60 ± 12.58	81.50 ± 8.59	61.73 ± 12.40	11.47%
Lym	148	18	**86.74 ± 8.11***	77.82 ± 10.01	80.54 ± 10.49	83.84 ± 10.57*	84.32 ± 9.59*	81.25 ± 8.21*	78.52 ± 8.70	2.87%
Hep	155	13	**74.70 ± 13.59***	64.05 ± 13.00	69.17 ± 15.65*	65.40 ± 13.25	61.73 ± 14.22	64.01 ± 14.89	65.65 ± 15.19	7.84%
Aud	200	69	**75.44 ± 7.60***	41.51 ± 7.20	36.70 ± 7.50	54.29 ± 8.96	64.83 ± 8.04	36.37 ± 7.60	58.55 ± 10.30	16.36%
Hsv	232	16	**96.65 ± 3.40**	94.28 ± 4.95	91.09 ± 5.55	95.81 ± 4.15	94.90 ± 4.14	91.59 ± 5.14	93.77 ± 5.30	0.88%
Spc	267	22	**53.09 ±10.35***	51.31 ± 9.16*	52.94 ± 9.48*	52.70 ± 9.69*	51.11 ± 8.97*	51.18 ± 7.90*	51.98 ± 8.85*	0.28%
Mof	300	10	**94.39 ±5.86***	79.35 ± 9.07	68.74 ± 10.58	79.35 ± 9.07	71.21 ± 8.42	77.70 ± 11.44	74.82 ± 8.08	18.95%
SoyL	307	35	90.97 ± 7.06	**93.45 ± 4.87***	64.92 ± 10.09	93.43 ± 4.95*	92.87 ± 5.35*	90.05 ± 4.92	89.84 ± 7.21	0.00%
Prim	339	17	**35.76 ± 8.61***	23.09 ± 6.71	27.00 ± 6.80	28.30 ± 7.93*	27.46 ± 7.56*	20.80 ± 6.64	29.42 ± 9.53	21.55%
Monk	432	6	**100 ± 0.00***	34.85 ± 0.00	99.88 ± 0.52*	34.85 ± 0.00	34.85 ± 0.00	**100 ± 0.00***	92.06 ± 5.24	0.00%
Tr	512	9	**91.01 ±2.93***	32.00 ± 0.00	75.88 ± 8.41	32.00 ± 0.00	32.00 ± 0.00	78.84 ± 5.09	78.84 ± 5.09	15.44%
Ba	625	4	**58.91 ±1.31***	21.25 ± 0.00	41.80 ± 5.82	21.25 ± 0.00	21.25 ± 0.00	39.32 ± 4.25	39.32 ± 4.25	40.93%
Crx	690	9	**83.26 ± 5.68***	78.58 ± 4.74	77.54 ± 5.68	82.79 ± 3.86*	81.02 ± 4.08	77.63 ± 5.12	78.28 ± 4.87	0.57%
Br	699	9	95.72 ± 2.07*	94.07 ± 2.84	93.44 ± 3.21	**96.34 ± 2.00***	94.14 ± 2.50	92.65 ± 3.42	93.93 ± 2.33	0.00%
Ma	830	4	**79.61 ±4.59***	70.22 ± 7.12*	70.14 ± 7.10*	70.20 ± 7.02*	70.22 ± 7.81*	69.79 ± 7.11 *	69.95 ± 7.29*	13.37%
Tic	958	9	92.80 ± 3.49	90.56 ± 2.70	78.29 ± 5.55	**100 ± 0.00***	89.72 ± 3.79	46.65 ± 4.10	46.56 ± 4.65	0.00%
Flr	1066	11	**59.88 ± 3.36***	57.01 ± 4.38*	57.11 ± 3.09	54.41 ± 3.39	55.61 ± 3.13	55.88 ± 4.38	54.98 ± 4.00	4.85%
Titn	2201	3	**23.33 ± 2.48***	10.54 ± 1.76	10.06 ± 0.62	10.06 ± 0.99	10.54 ± 1.76	10.54 ± 1.76	10.54 ± 1.76	32.48 %
DNAN	3186	60	**93.12 ± 1.05***	77.52 ± 1.21	52.22 ± 0.00	80.33 ± 1.48	91.65 ± 1.39	81.46 ± 1.75	69.11 ± 1.45	1.60 %
Spl	3190	60	**93.69 ± 1.11***	77.25 ± 2.19	24.45 ± 0.00	79.85 ± 2.07	84.96 ± 2.21	81.05 ± 1.81	69.29 ± 2.24	10.28 %
Krv	3196	36	**96.98 ± 1.06***	91.77 ± 1.66	90.04 ± 1.65	92.46 ± 1.74	91.39 ± 2.05	89.00 ± 1.43	91.48 ± 1.68	4.89%
Ld	3200	24	**63.37 ± 1.94***	62.11 ± 1.85*	41.35 ± 2.74	61.81 ± 1.98*	62.58 ± 1.85*	47.89 ± 2.37	41.57 ± 2.19	1.26 %
Ms	5644	22	**100 ± 0.00***	99.98 ± 0.06*	**100 ± 0.00***	**100 ± 0.00** *	**100 ± 0.00***	**100 ± 0.00** *	**100 ± 0.00***	0.00%
Krk	28056	6	**53.62 ± 1.71***	52.66 ± 0.78*	NA	52.50 ± 0.96*	52.57 ± 1.02*	39.05 ± 0.70	10.42 ± 0.10	1.82%
Adt	30162	8	**84.91 ± 0.86***	68.13 ± 1.12	NA	68.20 ± 1.07	68.16 ± 1.14	67.76 ± 1.04	68.01 ± 1.04	24.50%
Cnt	67557	42	**56.33 ± 0.78***	48.23 ± 0.73	NA	46.95 ± 0.49	46.65 ± 0.55	53.22 ± 0.73	45.81 ± 0.72	5.84%
Cens	299285	35	**68.93 ± 0.55***	66.88 ± 0.40	NA	67.47 ± 0.43	66.66 ± 0.42	66.96 ± 0.55	67.16 ± 0.37	2.64%
AR	-	-	**1.45****	4.27	4.87	3.73	4.00	4.72	4.68	2.28

calculates their ARs by the Friedman test and Bonferroni–Dunn test (Demšar 2006). The χ_F^2 of Friedman test is 45.92, associated with p-value $9.42e^{-9}$. This result indicates that the performance of all the compared methods is not equal. Further, the Bonferroni–Dunn test evaluates the CD between HELIC and other methods and shows the CD at p-value < 0.05 is 1.64. As shown in Table 6.1, HELIC achieves an overall AR of 1.45. Compared with the best state-of-the-art method (i.e., Ahmad, with an AR of 3.73), HELIC improves by 2.28, more than the CD value, and shows statistical significance. All the comparison results are shown in Figure 6.2, which reveals that HELIC is significantly ($p < 0.05$) better than all the compared distance measures.

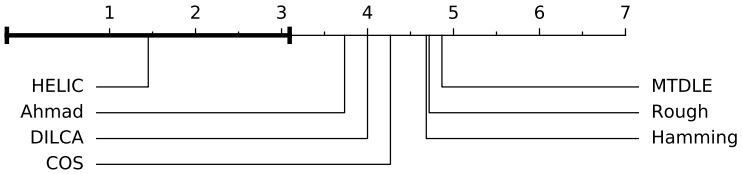
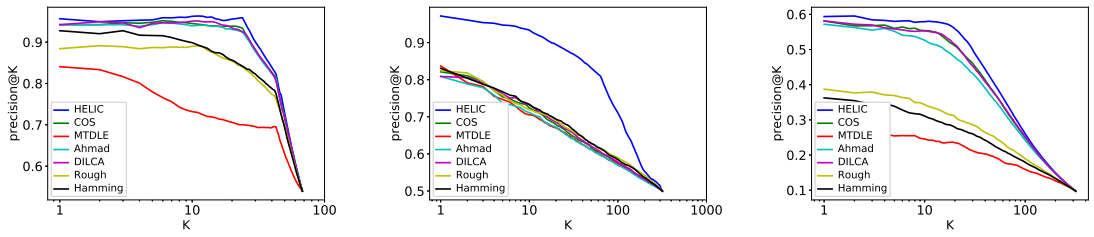


Figure 6.2: Comparison of HELIC against the other distance measures as per the Bonferroni–Dunn test. All distance measures with ranks outside the marked interval are significantly different (p < 0.05) from HELIC.



(a) Curve on Br data set. (b) Curve on Krv data set. (c) Curve on Ld data set.

Figure 6.3: The precision@k-curve of different distance measures: A better metric yields a higher curve in this figure.

6.4.2.2 HELIC-Enabled Retrieval Performance

Further study is to test the performance of HELIC representation in object retrieval. Three data sets (i.e., Br, Krv, and Ld) are tested, in which all the compared distance measures can achieve similar KNN classification accuracy for retrieval performance evaluation. The results are shown in Figure 6.3, and the precision of HELIC-enabled retrieval consistently outperforms the others. This high performance reflects that HELIC

can capture more details of distribution than can other distance measures. This ability of HELIC is powered by hierarchical coupling learning and heterogeneity learning.

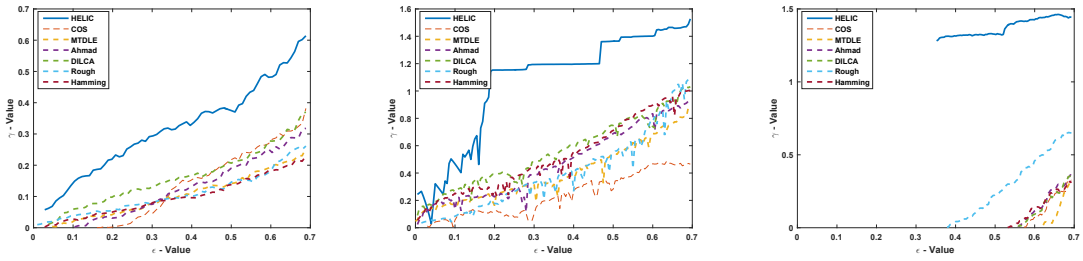
6.4.3 Testing HELIC Representation Quality

This experiment draws the (ϵ, γ)-curves on three data sets (i.e., DNAP, Aud, and Spc) to evaluate the representation quality of the proposed HELIC. To evaluate the performance of distance metric, it transforms the learned distance metric d to similarity measure s as follows:

$$s_{ij} = 1 - d_{ij}. \tag{6.40}$$

The transformed similarity s is then used for the (ϵ, γ)-good criterion.

The results are shown in Figure 6.4. It should be noted that this experiment focuses only on ϵ that can guarantee a non-negative margin (i.e., $\gamma \geqslant 0$). Therefore, Figure 6.4 displays only part of the (ϵ, γ)-curve, in which $\gamma \geqslant 0$.



(a) (ϵ, γ)-curve on DNAP data set. (b) (ϵ, γ)-curve on Aud data set. (c) (ϵ, γ)-curve on Spc data set.

Figure 6.4: The (ϵ, γ)-curve of different transformed similarity measures: A better metric would yield a curve with higher y-axis values.

The results illustrate that the proposed HELIC is better than its competitors in terms of (ϵ, γ)-good criterion. The results also reveal the insight behind the KNN classification performance in Table 6.1. For the DNAP and Aud data sets, the HELIC-enabled KNN has much higher F-score than the others, since HELIC yields a larger margin between different classes, which is reflected by the (ϵ, γ)-good in Figures 6.4(a) and 6.4(b). For the Spc data set, all methods have a low F-score, and the HELIC-enabled KNN is only slightly better than its competitors. It performs better because none of the methods can well distinguish nearly 40% part of the Spc data, but HELIC can separate more data than the others, as shown in Figure 6.4(c).

6.4.4 Testing the Effect of Learning Couplings and Heterogeneity

To evaluate the contribution of learning hierarchical couplings and heterogeneity, this thesis compares HELIC with its variant metric, hierarchical coupling (HC), which captures only hierarchical couplings. The HC method concatenates on the intra-attribute coupling space, inter-attribute coupling space and attribute-class coupling space for each attribute to construct a value-to-class coupling space. The Euclidean distance in this space is then used as the metric. Meanwhile, HELIC is compared with its variant HELIC-linear, which adopts only a linear kernel to learn a homogeneous metric in the hierarchical coupling space. The comparative classification results are shown in Table 6.2.

It can be seen from Table 6.2 that the overall ARs of HELIC, HC, and HELIC-linear are 1.27, 1.98, and 2.75, respectively. The Friedman test shows that χ_F^2 of these results is 36.76 associated with p-value $1.04e^{-8}$, which means the performance of these three methods differs significantly. The comparisons between HELIC and its variants in terms of AR are illustrated in Figure 6.5, which shows that HELIC significantly outperforms HC and HELIC-linear in terms of its CD value of 0.60 of the Bonferroni–Dunn test, with p-value < 0.05. The results demonstrate that heterogeneity learning contributes to an additional 0.75 to the AR of hierarchical couplings. In some data sets, the F-score improvement ratio of HELIC in terms of HC is very large (e.g., 92.02% on Titn and 37.38% on Aud). However, HELIC does not achieve better performance than HC over all data sets, showing that not all data sets involve strong heterogeneity.

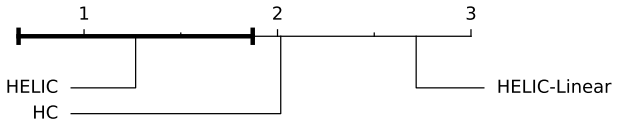


Figure 6.5: Comparison of HELIC against its variants per the Bonferroni–Dunn test. All distance measures with ranks outside the marked interval are significantly different (p < 0.05) from HELIC.

The results also demonstrate the significance of learning hierarchical couplings, since HC achieves the AR of 2.88 compared with the other six methods, as shown in Table 6.1. The AR of HC is better than that of the best state-of-the-art method, Ahmad (3.40). However, the HC performance does not differ significantly from that of others, as shown in Figure 6.6.

Table 6.2: KNN Classification F-score (%) with HELIC Variants. The Monte Carlo cross-validation results are reported as *mean* ± *standard deviation*. Δ shows the HELIC improvement over the best results of its variants.

Data set	HELIC	HC	HELIC-linear	Δ
SoyS	100 ± 0.00	100 ± 0.00	100 ± 0.00	0.00%
Zoo	100 ± 0.00	100 ± 0.00	100 ± 0.00	0%
DNAP	92.90±5.85	94.93 ± 7.00	85.77± 8.75	0%
Hay	90.85±5.07	85.89 ± 6.39	67.09 ± 13.94	5.77%
Lym	86.74 ± 8.11	77.69 ± 12.71	58.98 ± 15.96	11.65%
Hep	74.70 ± 13.59	70.08 ± 13.07	62.27 ± 15.30	6.65 %
Aud	75.44±7.60	54.94 ± 11.85	47.29 ± 7.24	37.31%
Hsv	96.65 ± 3.40	95.43 ± 4.46	94.48 ± 3.90	1.28%
Spc	53.09 ± 10.35	51.40 ± 9.51	50.15 ± 8.18	3.28%
Mfn	94.39 ± 5.86	94.92 ± 3.36	75.16 ± 10.24	0.00%
SoyL	90.97 ± 7.06	92.27 ± 3.86	89.72 ± 5.92	0.00%
Prim	35.76 ± 8.61	26.03 ± 5.82	24.71 ± 5.64	37.38%
Monk	100 ± 0.00	100 ± 0.00	100 ± 0.00	0.00%
Tr	91.01 ± 2.93	89.96 ± 2.92	76.23 ± 5.18	1.17%
Ba	58.91 ± 1.31	59.64 ± 1.46	44.99 ± 7.12	0%
Crx	83.26 ± 5.68	82.43 ± 4.39	81.34 ± 5.17	1.01%
Br	95.72 ± 2.07	94.19 ± 2.80	92.71 ± 2.67	1.62%
Ma	79.61 ± 4.59	70.31 ± 7.00	76.07 ± 8.45	4.65%
Tic	92.80 ± 3.49	79.73 ± 2.60	66.30 ± 4.65	16.39%
Flr	59.88 ± 3.36	55.40 ± 3.93	55.31 ± 4.32	8.09%
Titn	23.33 ± 2.48	12.15 ± 1.65	12.15 ± 1.65	92.02%
DNAN	93.12 ± 1.05	91.83 ± 1.64	87.43 ± 1.34	1.40%
Spc	93.69 ± 1.11	75.88 ± 2.03	63.45 ± 2.60	23.47%
Krv	96.98 ± 1.06	92.49 ± 0.92	95.27 ± 0.82	1.79%
Ld	63.37 ± 1.94	57.71 ± 2.46	51.56 ± 2.11	9.81%
Ms	100 ± 0.00	100 ± 0.00	100 ± 0.00	0.00%
Krk	53.62 ± 1.71	52.44 ± 1.58	52.03 ± 1.96	2.25%
Adt	84.91 ± 0.86	84.32 ± 0.80	68.16 ± 1.46	0.70%
Cnt	56.33 ± 0.78	43.07± 0.50	43.82 ± 0.67	28.55%
Cens	68.93 ± 0.55	64.23 ± 0.49	64.82 ± 0.52	7.32%
AR	**1.27****	2.02	2.72	0.75

The HELIC-linear performance is worse than that of HC, according to Table 6.2. These results indicate that the distributions in coupling spaces are complex and heterogeneous. Therefore, a learning metric in the linear space transformed by the linear kernel is insufficient. As analyzed in Section 6.3.2, a variety of kernels should be tested to capture the heterogeneity.

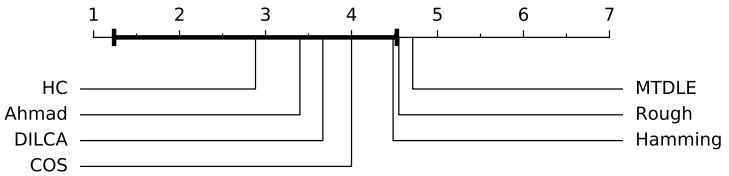


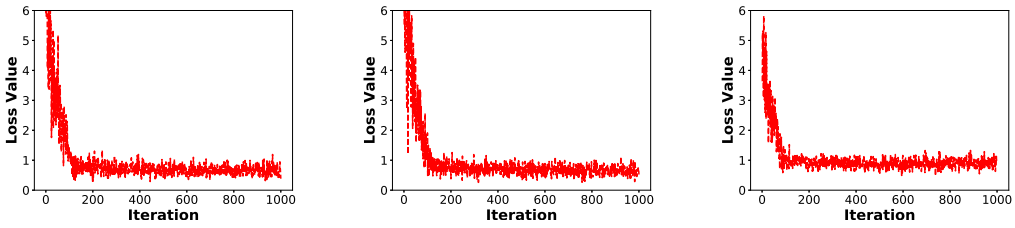
Figure 6.6: Comparison of HC against the other distance measures per the Bonferroni–Dunn test. All distance measures with ranks outside the marked interval are significantly different ($p < 0.05$) from HC.

6.4.5 Testing HELIC Scalability

The HELIC scalability is proportional to the number of iterations to achieve convergence, as discussed in Section 6.3.3. This section first empirically evaluates the convergence speed of HELIC and then illustrates HELIC's time cost under different data factors.

This experiment uses six real data sets to evaluate the HELIC convergence. These data sets represent data with a large number of attributes and objects. The method used to solve HELIC is Adam with the same settings as given in Section 6.4.1. The loss of objective function Equation (6.19) for these data sets is shown in Figure 6.7. The loss value converges rapidly within 200 iterations. This convergence is consistent with the theoretical analysis and demonstrates that HELIC time complexity is very low.

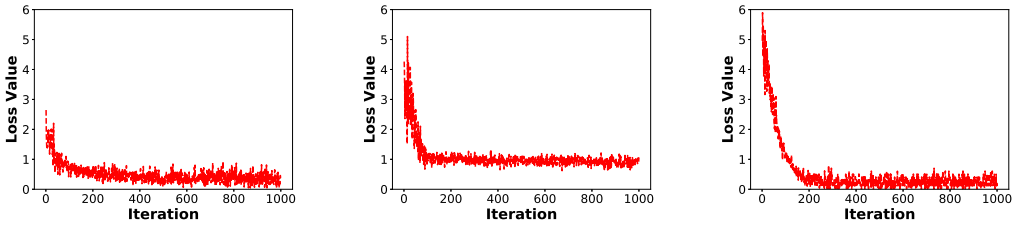
Further study is to generate synthetic data to test the computational cost of HELIC. The default data factors for synthetic data are as follows: the number of objects is $1,000$, the number of attributes is 10, and the maximum number of values in each attribute is 3. It generates three groups of data and tunes one of these factors for each group. For the first group of data, the number of objects is from $1,000$ to $100,000$. For the second group of data, the number of attributes is from 10 to 200. For the third group of data, the maximum number of values in the attributes is from 10 to 100. The HELIC time cost under each data factor is shown in Figure 6.8.



(a) Training loss on DNAN data set.

(b) Training loss on Spc data set.

(c) Training loss on Krv data set.



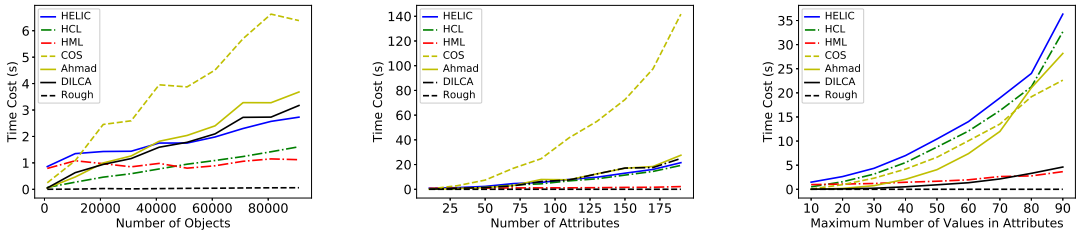
(d) Training loss on Ms data set.

(e) Training loss on Cnt data set.

(f) Training loss on Cens data set.

Figure 6.7: The HELIC training loss on different data sets. The stochastic optimization method for HELIC is Adam (Kingma & Ba 2014), the initial learning rate is 10^{-3}, and the batch size is 20. The x-axis refers to the number of iterations, and the y-axis refers to the loss value of HELIC metric learning objective function Equation (6.19).

As shown in Figure 6.8, the HELIC time cost is at the same level as that of most of the state-of-the-art methods. Although the Rough method has lower time complexity, it has a lower representation performance compared with others. Figure 6.8(a) shows that the HELIC time cost is almost stable (from 0.84 s to 3.47 s), which demonstrates its good scalability with respect to the amount of data n_o. Actually, the small time cost increase is related to Python's built-in functions when identifying a categorical value location in the data. Since this cost increases by an extremely small proportion with regard to the amount of data, it can be ignored when applying HELIC. Figure 6.8(b) and Figure 6.8(c) demonstrate that the time cost has a quadratic relation with both n_a and n_{mv}, which is consistent with the time complexity of HELIC, as analyzed in Section 6.3.3. These results also show that the main HELIC cost lies in hierarchical coupling learning (HCL), while the cost of heterogeneity and metric learning (HML) is constant. The reason for this constancy is that HELIC calculates the pairwise value relations when learning inter-attribute couplings. Hence, for categorical data with high dimensionality, the trade-off between capturing comprehensive complex couplings and preserving efficiency needs to be made.



(a) Time cost w.r.t. n_o. (b) Time cost w.r.t. n_a. (c) Time cost w.r.t. n_{mv}.

Figure 6.8: The HELIC time cost with respect to data factors: object number n_o, attribute number n_a, and maximum number of attribute values n_{mv}.

To evaluate the relation between time cost and the number of kernel functions, this section sets the number of kernels used in HELIC from 1 to 50 and tests the computational cost of HELIC on the synthetic data set with default data factors. The HELIC time cost with a different number of kernels is shown in Figure 6.9. This figure shows that the HELIC time cost is linear to the number of kernels, with a very small slope. Increasing the number of kernels only slightly affects the computational time of HELIC. This result is consistent with the theoretical analysis that indicates $n_{\boldsymbol{\omega}}$ is linear to the time complexity of HELIC. Here, $n_{\boldsymbol{\omega}}$ has a linear relation with the number of kernels.

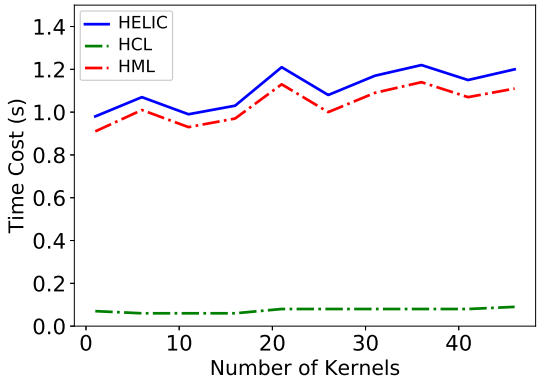


Figure 6.9: The HELIC time cost with respect to number of kernels.

6.4.6 Testing HELIC Stability

This experiment evaluates HELIC stability with regard to its parameter λ, which is a parameter that controls the weight of sparsity of $\boldsymbol{\omega}$. The larger λ is, the fewer components are selected for the construction of the finial metric, and vice versa. Figure 6.10

shows the HELIC-enabled KNN classification F-score under different settings of λ on the Krv data set. This result illustrates that HELIC is stable for a large range of λ, especially when λ is less than 1. However, the F-score drops rapidly when the value of λ surpasses 10, which indicates that the regularization dominates the objective function. In this case, the learned $\boldsymbol{\omega}$ cannot well reveal the heterogeneity and may even cause a loss of the learned couplings. Therefore, a small value for λ is recommended. A good choice for λ is $\frac{1}{n_k}$, which decreases as the size of $\boldsymbol{\omega}$ increases.

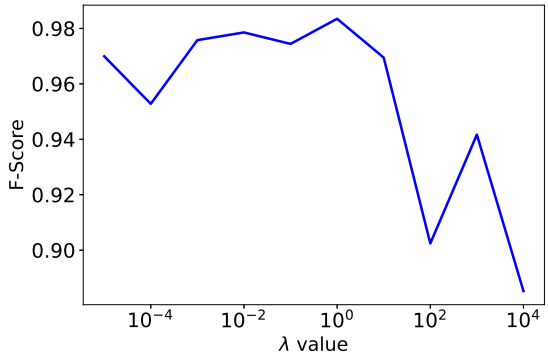


Figure 6.10: The HELIC-enabled KNN classification F-score with respect to λ.

6.5 Summary

This chapter studies how to jointly learn couplings and heterogeneities to form non-IID-completeness learning. It reports on an effective heterogeneous metric HELIC for learning hierarchical couplings within and between attributes and between attributes and classes in categorical data. The HELIC metric analyzes the heterogeneities in hierarchical interaction spaces and integrates heterogeneous couplings in complex categorical data. Both theoretical and experimental analyses show HELIC's effectiveness and efficiency in classifying categorical data with diverse data characteristics.

The research in this chapter is on supervised non-IID-completeness learning. It requires side information (e.g., class labels) to integrate heterogeneous dependencies and distributions. However, complex categorical data may lack such side information in lots of real-world applications, in which unsupervised non-IID-completeness learning is demanded. Furthermore, although this research proposes a method to integrate the redundancy and complementary information in heterogeneous couplings, the proposed method does not consider the inconsistency of these couplings, which may decrease the

integration performance as the number of captured heterogeneous couplings increases. To complement this chapter's research, the next chapter will study unsupervised non-IID-completeness learning and the method to mitigate the impact of the inconsistency.

CHAPTER 7

UNSUPERVISED CATEGORICAL REPRESENTATION WITH HETEROGENEOUS AND INCONSISTENT COUPLINGS

7.1 Introduction

The work in Chapter 6 effectively analyzes and captures the non-IID-completeness for supervised learning. Following this work, this chapter focuses on unsupervised non-IID-completeness learning for unlabeled categorical data representation. Furthermore, this chapter systematically studies how to effectively capture the coupling heterogeneity discussed in Section 2.4.3.

This chapter addresses the above research problems by introducing unsupervised heterogeneous coupling learning (UNTIE). This method simultaneously represents heterogeneous couplings with respect to (1) various value couplings, from attribute couplings to object couplings, (2) the complex relations between various couplings, and (3) heterogeneous distributions of various couplings by unsupervised kernel learning. Complex relations are entangled by the nonlinear mapping of various kernelized coupling functions, and the heterogeneous distributions are reflected by different kernels that are sensitive to different distributions. Instead of directly combining heterogeneous couplings, UNTIE first remodels various couplings into multiple kernels to transform the various couplings-based spaces into respective kernelized representation spaces with higher dimensionality. Then, UNTIE learns both the weight of each attribute value in

an individual kernel space and the weights of the learned kernel spaces to reflect both heterogeneous data distributions of respective couplings and interactions between couplings. Further, to efficiently learn heterogeneous couplings, UNTIE seamlessly wraps the kernel space weights with a positive semi-definite kernel and optimizes this kernel by optimizing an unsupervised kernel k-means objective. Lastly, the optimized kernel is used as the similarity representation of categorical data to generate a vector representation by further decomposing this kernel. This thesis provides theoretical analysis (see Theorem 7.3) that shows UNTIE represents categorical data with maximized separability for further learning tasks.

This chapter delivers the following significant contributions to categorical data representation:

- UNTIE is the very first unsupervised categorical data representation method to learn various value-to-object couplings and their complementarity and inconsistency. This method collectively captures heterogeneous data distributions of various couplings and adaptively integrates the couplings with their interactions.

- UNTIE maps various intra- and inter-attribute couplings into multiple kernels to capture the coupling heterogeneity (§7.2.3). In the kernel spaces, learning heterogeneous couplings is formalized as an efficient unsupervised optimization problem by optimizing an UNTIE-enabled kernel k-means objective (§7.2.4).

- UNTIE works in a completely unsupervised fashion to capture the intrinsic data characteristics in categorical data for amenable representations of categorical data complexity. Theoretical analysis shows that the UNTIE-represented data has the minimum normalized cut and increases data separability (§7.3).

The UNTIE method is evaluated by comprehensive experiments on 25 real-life categorical data sets with diversified data characteristics and four synthetic data sets generated as per a variety of data factors. The experimental results show that (1) UNTIE can effectively address both complementarity and inconsistency in learning heterogeneous couplings and (2) UNTIE enjoys an accuracy improvement (up to 51.72%, in terms of F-score on these data sets) from the learned heterogeneity and produces substantially better representation performance than the state-of-the-art categorical representation methods; (3) the efficiency of UNTIE is insensitive to the volume of data, which indicates UNTIE is scalable for large data; and (4) UNTIE can enhance various learning tasks.

The rest of this chapter is organized as follows. Section 7.2 outlines the UNTIE method and details its implementation. Section 7.3 conducts the theoretical analysis of the UNTIE properties. Section 7.4 evaluates the UNTIE performance in terms of different metrics. Lastly, Section 7.5 concludes this chapter and discusses prospects of future research.

7.2 The UNTIE Design

The proposed UNTIE aims to comprehensively and efficiently learn both complementary and inconsistent heterogeneous couplings. It learns couplings on each categorical value with respect to its multiple distributions, and it then combines the learned couplings in terms of the interactions between their distributions. The rationale behind UNTIE includes the following propositions: (1) A categorical value may belong to multiple distributions. (2) A coupling may make different contributions to different value distributions. (3) The overall distribution of a categorical value can be described by multiple distributions. The above are called *heterogeneity hypotheses*, which are theoretically supported by Theorem 7.1 in Section 7.3.1.

7.2.1 The UNTIE Framework

While this chapter presents a specific instance of UNTIE, as shown in Figure 7.1, UNTIE actually represents a framework of unsupervised categorical data representation. It represents categorical data in both vector (as a vector representation) and similarity (as a kernel matrix) spaces. To reveal heterogeneous couplings, UNTIE first converts categorical data to several coupling spaces by multiple coupling learning functions. It then feeds and transforms each coupling space into multiple kernel spaces. Afterward, it reduces the redundancy and inconsistency between heterogeneous couplings by learning the heterogeneity between couplings in the kernel spaces. Specifically, UNTIE differentiates the contributions of individual kernel spaces and reveals the kernel-sensitive distribution within each kernel space. To efficiently learn the heterogeneity in an unsupervised way, UNTIE wraps the weight of each kernel space and the weight of the values embedded in a kernel space by a wrapper kernel, and then it optimizes this kernel by solving a kernel k-means objective (i.e., regularizing the objects within one cluster to be more similar to each other than to those in other clusters). The UNTIE method uses the optimized wrapper kernel as the similarity representation of categorical data. It further

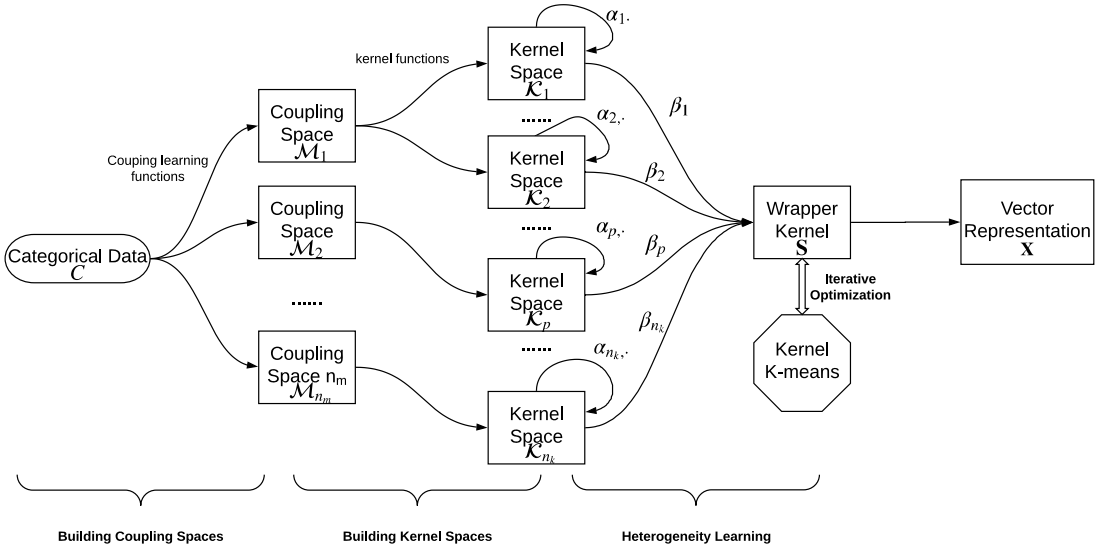


Figure 7.1: The UNTIE framework: It first transforms the coupling spaces into multiple kernel spaces and then learns the heterogeneity within and between couplings in these kernel spaces by solving a kernel k-means objective.

generates the vector representation by decomposing the optimized wrapper kernel.

To effectively address the coupling inconsistency problem, UNTIE learns heterogeneous couplings by multiple kernels, which transforms a coupling from its original space to several kernel spaces. Since a kernel is sensitive to a distribution (Bucak et al. 2014), these kernel spaces reflect different value distributions. In a kernel space, a coupling is preserved if it matches the kernel-sensitive distribution while other couplings are filtered. For this purpose, UNTIE learns the weights of values in each kernel space to effectively reveal the multiple distributions corresponding to a coupling. The multiple distributions of a categorical value jointly contribute to the overall distribution of the value. Therefore, the learned weights of these kernel spaces filter the redundant information but integrate the complementary information.

To learn the heterogeneous couplings in an unsupervised manner, without loss of generality, UNTIE takes the assumption that the objects within one cluster are more similar to each other than to those in other clusters. This assumption is commonly made in most of clustering and classification methods and shows its validity with real data distributions. Accordingly, UNTIE iteratively learns heterogeneous couplings for categorical data representation. In each iteration, UNTIE first analyzes the clusters based on its generated representation, and it then tunes the representation based on the

obtained clusters. To efficiently cluster data, UNTIE wraps the weight of each kernel space and the weight of the values embedded in kernel spaces by a wrapper kernel, and optimizes this kernel by solving a kernel k-means objective. In addition to efficiency, the kernel k-means objective also brings other benefits for categorical data representation, such as good separability, as will be discussed in Section 7.3.3.

7.2.2 Heterogeneous Coupling Learning

As mentioned before, this thesis broadly refers to *couplings* as any interactions and relations between values, attributes, and objects (Cao 2015, Wang, Dong, Zhou, Cao & Chi 2015, Zhu et al. 2018). In categorical data, possible types of couplings include *intra-attribute couplings* (e.g., value couplings, such as value frequency, co-occurrence, and matching), inter-attribute couplings (e.g., attribute correlation and dependency), and object couplings built on value and attribute couplings. The UNTIE method learns value-to-attribute-to-object hierarchical couplings on top of capturing and fusing various intra- and inter-attribute couplings in unlabeled categorical data.

Learning Intra-Attribute Couplings. *Intra-attribute couplings* represent the interactions between the values of an attribute and the value distributions in an attribute (Boriah et al. 2008, Wang, Chi, Zhou & Wong 2015). The UNTIE method measures intra-attribute couplings in terms of the intra-attribute distributions by a value frequency function and calculates the Euclidean distance in a numerical space. Although the value frequency function has only one input value, it measures the value distribution against all values. For a categorical value $\mathsf{v}_i^{(j)}$ in the j-th attribute, the value frequency function $m_{Ia}^{(j)}(\mathsf{v}_i^{(j)})$ maps an intra-attribute coupling between this value and the other categorical values in this attribute to a one-dimensional intra-attribute coupling vector $m_{Ia}^{(j)}(\mathsf{v}_i^{(j)})$ as follows:

$$m_{Ia}^{(j)}(\mathsf{v}_i^{(j)}) = [\frac{|g^{(j)}(\mathsf{v}_i^{(j)})|}{n_o}], \tag{7.1}$$

where $g^{(j)}(\cdot) : V^{(j)} \rightarrow O$ maps the value $\mathsf{v}_i^{(j)}$ to a set of objects that have value $\mathsf{v}_i^{(j)}$ in the j-th attribute, n_o is the number of objects, and $|\cdot|$ refers to the count of a set. For example, in Table 2.1, a relationship between value *yellow* and object attribute *color* is $g^{(j)}(yellow) = \{A2, A3\}$. The intra-attribute coupling vector of value *yellow* is $m_{Ia}^{(2)}(yellow) = [\frac{|\{A2,A3\}|}{6}] = [\frac{1}{3}]$.

An intra-attribute coupling space $\mathscr{M}_{Ia}^{(j)}$ is spanned by the intra-attribute coupling

vectors obtained in an attribute by Equation (7.1) and is defined as follows:

$$\mathcal{M}_{Ia}^{(j)} = \{m_{Ia}^{(j)}(\mathsf{v}_i^{(j)}) | \mathsf{v}_i^{(j)} \in V_j\}. \tag{7.2}$$

For categorical data with n_a attributes, the intra-attribute coupling spaces $\mathcal{M}_{Ia}$ are constructed as follows:

$$\mathcal{M}_{Ia} = \{\mathcal{M}_{Ia}^{(1)}, \cdots, \mathcal{M}_{Ia}^{(n_a)}\}. \tag{7.3}$$

The above intra-attribute coupling spaces present only a one-dimensional embedding of the categorical data space with respect to each attribute, which does not consider the interactions between attributes. Accordingly, the following inter-attribute couplings complement the intra-attribute couplings.

Learning Inter-Attribute Couplings. *Inter-attribute couplings* refer to the interactions between attributes and the contextual (or semantic) information of attribute values with respect to other attributes (Ienco et al. 2012). This attribute-based interactive and contextual information complements the value distributions captured in intra-attribute couplings. For example, in Table 2.1, white and black watermelons have the same frequency but can be distinguished by involving their *root shapes*, which are differ significantly.

Here, the inter-attribute couplings are represented by the information conditional probability, which reveals the distributions of an attribute value in the spaces spanned by the values of the other attributes. Given a value $\mathsf{v}^{(j)}$ of attribute a_j and a value $\mathsf{v}^{(k)}$ of attribute a_k, the information conditional probability function is defined as follows:

$$p(\mathsf{v}^{(j)} | \mathsf{v}^{(k)}) = \frac{|g^{(j)}(\mathsf{v}^{(j)}) \cap g^{(k)}(\mathsf{v}^{(k)})|}{|g^{(k)}(\mathsf{v}^{(k)})|}, \tag{7.4}$$

where $\cap$ returns the intersection of two sets. Based on the information conditional probability function, the inter-attribute coupling learning function $m_{Ie}^{(j)}(\mathsf{v}_\mathsf{i}^{(j)})$ embeds interactions between value $\mathsf{v}_\mathsf{i}^{(j)}$ and other attributes as a $|V_*|$-dimensional inter-attribute coupling vector $m_{Ie}^{(j)}(\mathsf{v}_i^{(j)})$,

$$m_{Ie}^{(j)}(\mathsf{v}_i^{(j)}) = \left[\begin{array}{cccc} p(\mathsf{v}_i^{(j)} | \mathsf{v}_{*1}), & \cdots, & p(\mathsf{v}_i^{(j)} | \mathsf{v}_{*|V_*|}) \end{array} \right]^\top, \tag{7.5}$$

where $V_* = \{V^{(k)} | k \in N_a, k \neq j\}$ is the set of values in all attributes except a_j and where $\mathsf{v}_{*i} \in V_*$ is a categorical value in set V_*. For example, the information condition probability between the yellow watermelons and those with curled root shape is $p(yellow|curled) =$

$\frac{|\{A2,A3\}\cap\{A3,A5\}|}{|\{A3,A5\}|} = \frac{|\{A3\}|}{|\{A3,A5\}|} = \frac{1}{2}$. The inter-attribute coupling vector of value *yellow* is calculated as

$$\begin{aligned} & m_{Ie}^{(2)}(yellow) \\ &= [p(yellow|clear), \cdots, p(yellow|slightly\ curled)] \\ &= [0, \frac{2}{3}, \frac{1}{2}, \frac{1}{2}, 0] \end{aligned}$$

An inter-attribute coupling space $\mathcal{M}_{Ie}^{(j)}$ is spanned by the inter-attribute coupling vectors obtained in an attribute by Equation (7.5):

$$\mathcal{M}_{Ie}^{(j)} = \{m_{Ie}^{(j)}(\mathsf{v}_i^{(j)})|\mathsf{v}_i^{(j)} \in V^{(j)}\}. \tag{7.6}$$

For categorical data with n_a attributes, the inter-attribute coupling spaces $\mathcal{M}_{Ie}$ are calculated as follows:

$$\mathcal{M}_{Ie} = \{\mathcal{M}_{Ie}^{(1)}, \cdots, \mathcal{M}_{Ie}^{(n_a)}\}. \tag{7.7}$$

If $|V| > 2|V^{(j)}| - 1$, an inter-attribute coupling learning function would project categorical values into a higher dimensional space, because the dimensionality of inter-attribute coupling space equals $|V| - |V^{(j)}|$, while the degree of freedom (equivalent to dimensionality of transforming a categorical value to a dummy variable) of the j-th attribute is $|V^{(j)}| - 1$. In this way, the value couplings affected by other attributes are captured, which complements with the intra-attribute couplings to form a complete representation of categorical attribute space.

7.2.3 Heterogeneity Learning in Kernel Spaces

With the coupling spaces built above from intra- and inter-perspectives, UNTIE further constructs an entire coupling space $\mathcal{M}$, which is a collection of heterogeneous intra-attribute coupling spaces $\mathcal{M}_{Ia}$ and inter-attribute coupling spaces $\mathcal{M}_{Ie}$:

$$\mathcal{M} = \mathcal{M}_{Ia} \cup \mathcal{M}_{Ie}. \tag{7.8}$$

To effectively integrate the heterogeneous couplings in the learned coupling space set, UNTIE transforms the learned heterogeneous coupling spaces into uniform spaces, in which heterogeneous couplings are comparable. Specifically, UNTIE uses multiple kernels to transform each coupling space into its corresponding kernel spaces, where each kernel space corresponds to the transformed coupling space with respect to a particular kernel mapping function. It generates a set of n_k ($n_k = |\mathcal{M}| \times |F|$, where F is

the set of kernel functions for the transformation) kernel spaces $\{\mathcal{K}_1, \mathcal{K}_2, \cdots, \mathcal{K}_{n_k}\}$, and the p-th ($p \leq n_k$) space is spanned by a kernel matrix $\mathbf{K}_p$, which is constructed from a coupling space $\mathcal{M}_j$ by a kernel function $k_p(\cdot,\cdot)$ for an attribute. Denoting m_i as a vector in $\mathcal{M}_j$ corresponding to the i-th categorical value, $\mathbf{K}_p$ is represented as follows:

$$\mathbf{K}_p = \begin{bmatrix} k_p(\mathsf{m}_1, \mathsf{m}_1) & k_p(\mathsf{m}_1, \mathsf{m}_2) & \cdots & k_p(\mathsf{m}_1, \mathsf{m}_{n_v^*}) \\ k_p(\mathsf{m}_2, \mathsf{m}_1) & k_p(\mathsf{m}_2, \mathsf{m}_2) & \cdots & k_p(\mathsf{m}_2, \mathsf{m}_{n_v^*}) \\ \vdots & \vdots & \ddots & \vdots \\ k_p(\mathsf{m}_{n_v^*}, \mathsf{m}_1) & k_p(\mathsf{m}_{n_v^*}, \mathsf{m}_2) & \cdots & k_p(\mathsf{m}_{n_v^*}, \mathsf{m}_{n_v^*}) \end{bmatrix}, \tag{7.9}$$

where n_v^* is the number of categorical values represented by $\mathcal{M}_j$. For example, in Table 2.1, if $k_p(\cdot,\cdot)$ is the linear kernel and $\mathcal{M}_j$ is $\mathcal{M}_{Ie}^{(2)}$, let m_2 correspond to value $yellow$, $k_p(\mathsf{m}_2, \mathsf{m}_2) = [0, \frac{2}{3}, \frac{1}{2}, \frac{1}{2}, 0]^\top \cdot [0, \frac{2}{3}, \frac{1}{2}, \frac{1}{2}, 0] = \frac{17}{18}$.

To reveal the heterogeneity within a coupling, UNTIE learns the weights of values in each kernel space. Specifically, it learns a set of transformation matrices $\{\mathbf{T}_1, \mathbf{T}_2, \cdots, \mathbf{T}_{n_k}\}$ to reconstruct the kernel spaces $\{\mathcal{K}'_1, \cdots, \mathcal{K}'_{n_k}\}$, in which the p-th kernel matrix $\mathbf{K}'_p$ contains only the p-th kernel sensitive distribution that is suited for the corresponding coupling. The reconstructed kernel spaces are called *heterogeneous kernel spaces*. Kernel matrix $\mathbf{K}'_p$ is defined as follows:

$$\mathbf{K}'_p = \mathbf{T}_p \cdot \mathbf{K}_p. \tag{7.10}$$

The UNTIE method regulates $\mathbf{T}_p$ as a diagonal matrix:

$$\mathbf{T}_p = \begin{bmatrix} \alpha_{p1} & 0 & \cdots & 0 \\ 0 & \alpha_{p2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{pn_v} \end{bmatrix}. \tag{7.11}$$

As a result, α_{pi} is the weight of the i-th value in the p-th kernel space, that is,

$$[k_p(\mathsf{m}_1, \mathsf{m}_1), k_p(\mathsf{m}_1, \mathsf{m}_2), \cdots, k_p(\mathsf{m}_1, \mathsf{m}_{\mathsf{n}_\mathsf{v}^*})].$$

The larger α_{pi} implies stronger coupling of the i-th value that is revealed by the coupling space corresponding to the p-th kernel space.

To further capture the heterogeneity between couplings, UNTIE learns the contribution of each heterogeneous kernel space to a final representation. Specifically, it first defines a similarity measure between objects in the heterogeneous kernel space, and it then learns the weight of each kernel space based on this similarity measure to reflect

their contribution. Given a categorical data set, considering the p-th kernel matrix, let $\mathfrak{i}$ and $\mathfrak{j}$ represent the indices of values in the p-th kernel space corresponding to the i-th and j-th objects, respectively. Using $\mathbf{K}'_{p,\mathfrak{i}\cdot}$ (i.e., the $\mathfrak{i}$-th row in $\mathbf{K}'_p$) to denote the $\mathfrak{o}_i$ in the p-th heterogeneous kernel space, the similarity $S_{p,ij}$ measured by the linear kernel of the i-th and j-th objects in this space is as follows:

$$S_{p,ij} = \mathbf{K}'^{\top}_{p,\mathfrak{i}\cdot}\mathbf{K}'_{p,\mathfrak{j}\cdot}. \tag{7.12}$$

By considering Equation (7.10), Equation (7.12) equals

$$S_{p,ij} = \mathbf{K}^{\top}_{p,\mathfrak{i}\cdot}\mathbf{T}_p^{\top}\mathbf{T}_p\mathbf{K}_{p,\mathfrak{j}\cdot}. \tag{7.13}$$

The UNTIE method defines the final similarity representation S_{ij} between the i-th and j-th objects as a linear combination of base similarity measures from heterogeneous spaces to filter redundant information and integrate complementary information between couplings:

$$S_{ij} = \sum_{p=1}^{n_k} \beta_p S_{p,ij}, \tag{7.14}$$

where $\beta_p \geq 0$ is the weight for the p-th base similarity. Denoting a diagonal matrix $\boldsymbol{\omega}_p = \beta_p \mathbf{T}^{\top}\mathbf{T}$, Equation (7.14) is rewritten:

$$S_{ij} = \sum_{p=1}^{n_k} \mathbf{K}^{\top}_{p,\mathfrak{i}\cdot}\boldsymbol{\omega}_p\mathbf{K}_{p,\mathfrak{j}\cdot}. \tag{7.15}$$

In this way, UNTIE simultaneously learns α and β by learning $\boldsymbol{\omega}$, namely a heterogeneity parameter. The optimized $\boldsymbol{\omega}$ guides UNTIE to integrate the heterogeneous couplings into the similarity representation S_{ij}.

7.2.4 Kernel *K*-Means-Based Representation Learning

In an unsupervised way, UNTIE learns the heterogeneous couplings by wrapping α, which reveals the heterogeneity within couplings, and β, which reveals the heterogeneity between couplings, into a wrapper kernel, and further optimizes this kernel by solving a kernel k-means objective (Dhillon et al. 2004).

A popular clustering algorithm, k-means minimizes the distance between an object and its assigned cluster center, which is also used for information integration (Yu et al.

2012). Given a set of objects $O = \{\mathsf{o}_i \in \mathscr{R}^{n_a} | i = 1, \cdots, n_o\}$, the k-means objective is formalized as follows:

$$\begin{aligned} \underset{\mathbf{Z} \in \{0,1\}^{n_o \times n_c}}{\text{minimize}} \quad & \sum_{i=1,c=1}^{n_o,n_c} z_{ic} \|\mathsf{o}_i - \boldsymbol{\mu}_c\|_2^2 \\ \text{subject to} \quad & \sum_{c=1}^{n_c} z_{ic} = 1, \end{aligned} \tag{7.16}$$

where z_{ic} indicates whether o_i belongs to the c-th cluster, $\boldsymbol{\mu}_c = \frac{1}{n_{oc}} \sum_{i=1}^{n_o} z_{ic} \mathbf{x}_i$ is the centroid of the c-th cluster, and $n_{oc} = \sum_{i=1}^{n_o} z_{ic}$ refers to the size of the c-th cluster.

To address the issue that k-means cannot cluster data with a nonlinear boundary, the kernel k-means first uses a mapping function to map data to a higher dimensional space and then adopts k-means to cluster the mapped data. With a kernel function $k(\cdot)$, the kernel k-means is formalized as follows:

$$\begin{aligned} \underset{\mathbf{Z} \in \{0,1\}^{n_o \times n_c}}{\text{minimize}} \quad & \sum_{i=1,c=1}^{n_o,n_c} z_{ic} \|k(\mathsf{o}_i) - \boldsymbol{\mu}_c\|_2^2 \\ \text{subject to} \quad & \sum_{c=1}^{n_c} z_{ic} = 1, \end{aligned} \tag{7.17}$$

where $\boldsymbol{\mu}_c = \frac{1}{n_{oc}} \sum_{i=1}^{n_o} z_{ic} k(\mathsf{o}_i)$.

Equation (7.17) is rewritten:

$$\begin{aligned} \underset{\mathbf{Z} \in \{0,1\}^{n_o \times n_c}}{\text{minimize}} \quad & \text{Tr}(\mathbf{K}) - \text{Tr}(\mathbf{L}^{\frac{1}{2}} \mathbf{Z}^\top \mathbf{K} \mathbf{Z} \mathbf{L}^{\frac{1}{2}}) \\ \text{subject to} \quad & \mathbf{Z}\mathbf{1}_{n_c} = \mathbf{1}_{n_o}, \end{aligned} \tag{7.18}$$

where $\text{Tr}(\cdot)$ calculates the trace of a matrix, $\mathbf{K}$ is a matrix with $k_{ij} = k(\mathsf{o}_i^\top) k(\mathsf{o}_j)$, $\mathbf{L} = \text{diag}([n_{o1}^{-1}, n_{o2}^{-1}, \cdots, n_{on_c}^{-1}])$, and $\mathbf{1}_\ell \in \{1\}^\ell$ is a column vector with all elements being 1.

Directly solving Equation (7.18) is difficult, since the values of $\mathbf{Z}$ are limited to either 0 or 1. Typically, Equation (7.18) is relaxed by letting $\mathbf{Z}$ take real values. Denoting $\mathbf{H} = \mathbf{Z}\mathbf{L}^{\frac{1}{2}}$, the above problem is restated as

$$\begin{aligned} \underset{\mathbf{H}}{\text{minimize}} \quad & \text{Tr}(\mathbf{K}(\mathbf{I}_{n_o} - \mathbf{H}\mathbf{H}^\top)) \\ \text{subject to} \quad & \mathbf{H} \in \mathscr{R}^{n_o \times n_c}, \\ & \mathbf{H}^\top \mathbf{H} = \mathbf{I}_{n_c}, \end{aligned} \tag{7.19}$$

where $\mathbf{I}_{n_c}$ is an identity matrix with size $n_c \times n_c$. The optimal $\mathbf{H}$ for Equation (7.19) can be obtained by taking the n_c eigenvectors having large eigenvalues of $\mathbf{K}$ (Jegelka et al. 2009).

The UNTIE method integrates heterogeneous coupling learning into the kernel k-means seamlessly by wrapping α and β to a wrapper kernel $s(\cdot,\cdot): O \times O \to \mathscr{R}$, which is defined below:

$$s(\mathsf{o}_i, \mathsf{o}_j) = S_{ij}. \tag{7.20}$$

Accordingly, UNTIE constructs a kernel matrix $\mathbf{S}$ with respect to kernel $s(\cdot,\cdot)$ and categorical object set O:

$$\mathbf{S} = \begin{bmatrix} s(\mathsf{o}_1,\mathsf{o}_1) & s(\mathsf{o}_1,\mathsf{o}_2) & \cdots & s(\mathsf{o}_1,\mathsf{o}_{n_o}) \\ s(\mathsf{o}_2,\mathsf{o}_1) & s(\mathsf{o}_2,\mathsf{o}_2) & \cdots & s(\mathsf{o}_2,\mathsf{o}_{n_o}) \\ \vdots & \vdots & \ddots & \vdots \\ s(\mathsf{o}_{n_o},\mathsf{o}_1) & s(\mathsf{o}_{n_o},\mathsf{o}_2) & \cdots & s(\mathsf{o}_{n_o},\mathsf{o}_{n_o}) \end{bmatrix}. \tag{7.21}$$

Since $s(\cdot,\cdot)$ is proved as a valid positive semi-definite kernel (see details in Section 7.3.2), $\mathbf{S}$ can replace $\mathbf{K}$ in Equation (7.19). In this way, the objective function of kernel k-means-based representation learning can be formalized:

$$\begin{aligned} \underset{\mathbf{H},\boldsymbol{\omega}}{\text{minimize}} \quad & \text{Tr}(\mathbf{S}(\mathbf{I}_{n_o} - \mathbf{H}\mathbf{H}^\top)) \\ \text{subject to} \quad & \mathbf{H} \in \mathscr{R}^{n_o \times n_c}, \\ & \mathbf{H}^\top \mathbf{H} = \mathbf{I}_{n_c}, \end{aligned} \tag{7.22}$$

where $\boldsymbol{\omega}$ is a heterogeneity parameter to learn, and UNTIE obtains the similarity representation of categorical data as $\mathbf{S}$. The corresponding vector representation can be obtained by

$$\mathbf{x}_i = [\sqrt{\boldsymbol{\omega}_{1,11}}\mathbf{K}_{1,\mathsf{i}1}, \sqrt{\boldsymbol{\omega}_{1,22}}\mathbf{K}_{1,\mathsf{i}2}, \cdots, \sqrt{\boldsymbol{\omega}_{n_k,n_v^* n_v^*}}\mathbf{K}_{n_k,\mathsf{i}n_v^*}], \tag{7.23}$$

where $\boldsymbol{\omega}_{i,jj}$ refers to the value of the (j,j)-th entry in $\boldsymbol{\omega}_i$ and where n_v^* refers to the number of values in the attribute corresponding to the n_k-th kernel. The learned representation $\mathbf{x}_i$ is a numerical approximation of categorical data, which can be fed into vector-based learning methods.

7.2.5 The UNTIE Algorithm

The UNTIE objective function in Equation (7.22) can be solved by alternatively updating $\mathbf{H}$ and $\boldsymbol{\omega}$: (1) **Optimizing H given $\boldsymbol{\omega}$**: by fixing the parameter $\boldsymbol{\omega}$, $\mathbf{H}$ can be obtained by solving a kernel k-means clustering optimization problem shown in Equation (7.19)

by eigenvalue decomposition; (2) **Optimizing $\boldsymbol{\omega}$ given H**: with H fixed, the objective function of learning $\boldsymbol{\omega}$ is,

$$\underset{\boldsymbol{\omega}}{\text{minimize}} \quad \text{Tr}(\mathbf{S}(\mathbf{I}_{n_o} - \mathbf{H}\mathbf{H}^\top)), \tag{7.24}$$

which can be optimized by linear programming. For large-scale data, Equation (7.24) can be solved by the stochastic gradient descent (SGD) method (e.g., Duchi et al., 2011; Kingma & Ba, 2014). The computational cost of UNTIE with respect to different optimization methods will be analyzed in Section 7.3.5. Algorithm 3 explains the UNTIE working process.

Algorithm 3 The UNTIE Algorithm for Unsupervised Representation Learning

Require: Categorical data set C, a set of kernel functions $K = \{k_1(\cdot,\cdot), \cdots, k_{n_k^*}(\cdot,\cdot)\}$, the number of clusters n_c, and convergence rate δ.
Ensure: Similarity representation $\mathbf{S}$, vector representation $\mathbf{X}$.
1: Mapping categorical data to coupling spaces according to Equations (7.2) and (7.6).
2: Mapping coupling spaces to multiple kernel spaces $\{\mathbf{K}_1, \cdots, \mathbf{K}_{n_k}\}$ by using K according to Equation (7.9).
3: Initializing the wrapper kernel matrix $\mathbf{S}$ by setting α and β as 1, and setting $l' = +\infty$ and $\Delta = +\infty$.
4: **for** $\Delta > \delta$ **do**
5: Calculating the n_c eigenvectors that have the largest eigenvalues of $\mathbf{S}$. Constructing $\mathbf{H}$ by these eigenvectors.
6: Optimizing $\boldsymbol{\omega}$ by solving Equation (7.24).
7: Calculating loss per $l = \text{Tr}(\mathbf{S}(\mathbf{I}_{n_o} - \mathbf{H}\mathbf{H}^\top))$.
8: Calculating loss change per $\Delta = |l - l'|$
9: Setting $l' = l$.
10: $n_i = n_i + 1$.
11: **end for**
12: Calculating the vector representation $\mathbf{X}$ per Equation (7.23).
13: **return** $\mathbf{S}$, $\mathbf{X}$

7.3 Theoretical Analysis of UNTIE Properties

7.3.1 The Fitness of Heterogeneity Hypotheses

To discuss the fitness of heterogeneity hypotheses, the following theorem is introduced first.

Theorem 7.1. *The distribution of a categorical data set, Φ, can be described as a probability tensor Φ, where each entry corresponds to the joint probability of a set of categorical*

values from n_a different attributes. Tensor Φ can be decomposed as $\Phi = \sum_{h=1}^{k} \pi_h \Theta_h$, where $\Theta_h = \boldsymbol{\theta}_h^{(1)} \otimes \boldsymbol{\theta}_h^{(2)} \otimes \cdots \otimes \boldsymbol{\theta}_h^{(n_a)}$, π_h is the weight of Θ_h for composing Φ, $\otimes$ refers to the outer product, and $\boldsymbol{\theta}_h^{(j)}$ is a probability vector of categorical values in the j-th attribute with a size of $n_v^{(j)} \times 1$ for $h = 1, \cdots, k$ and $j = 1, \cdots, n_a$.

Theorem 7.1 can be proved by Corollary 1 in Dunson & Xing (2009). The categorical data distribution Φ is a joint distribution of categorical values in each attribute. It is defined as $\Phi = \{\phi_{v^{(1)}v^{(2)}\cdots v^{(n_a)}} | v^{(j)} \in V^{(j)}\}$, where $\phi_{v^{(1)}v^{(2)}\cdots v^{(n_a)}}$ are the probabilities of values $v^{(1)}, v^{(2)}, \cdots, v^{(n_a)}$ co-occur. Tensor Φ is a probability tensor for which each entry is an element in Φ.

Theorem 7.1 indicates the following categorical data characteristics:

- *A value may belong to multiple distributions*. For different h values in Theorem 7.1, a categorical value in a_j will have different distributions $\boldsymbol{\theta}_h^{(j)}$. Therefore, if $h > 1$, the categorical value may have different distributions.

- *A coupling may contribute differently to different distributions*. The interactions under various distributions may differ. For distribution Θ_h, the attributes are independent (indicated by Θ_h, which equals the outer product of attribute distributions). In this case, inter-attribute couplings may not contribute. On the contrary, for distribution Φ, the attributes interact with each other, as is mainly reflected by inter-attribute couplings.

- *Overall distribution of a categorical value is a mixture of multiple distributions*. In Theorem 7.1, the overall distribution of a categorical value equals the weighted sum of its multiple distributions. The parameter π_h reflects the interactions between the distributions.

The heterogeneity hypotheses fit the above categorical data characteristics and provide a solid foundation for UNTIE to effectively capture the heterogeneity in couplings.

7.3.2 The Positive Semi-Definite Property of UNTIE Wrapper Kernel

As stated in Section 7.2.4, $s(\cdot,\cdot)$ has to be a positive semi-definite kernel to enable that $\mathbf{S}$ can be integrated into kernel k-means objective Equation (7.22). Before the above is proved, a lemma of kernel properties is introduced.

Lemma 7.1. *If $k_1(\mathrm{o}_i,\mathrm{o}_j)$ and $k_2(\mathrm{o}_i,\mathrm{o}_j)$ are positive semi-definite kernels in $O\times O$, and a constant $a>0$, then the following $k(\mathrm{o}_i,\mathrm{o}_j)$ functions are positive semi-definite kernels:*

(1) $k(\mathrm{o}_i,\mathrm{o}_j)=ak_1(\mathrm{o}_i,\mathrm{o}_j)$ and

(2) $k(\mathrm{o}_i,\mathrm{o}_j)=k_1(\mathrm{o}_i,\mathrm{o}_j)+k_2(\mathrm{o}_i,\mathrm{o}_j)$.

This lemma can be found in Section 4.1 of Steinwart & Christmann (2008).

Theorem 7.2. *The wrapper kernel $s(\mathrm{o}_i,\mathrm{o}_j)=S_{ij}$, which is defined in Equation* (7.20)*, is a positive semi-definite kernel.*

Proof. Given coupling spaces and multiple kernel functions, the i-th object o_i corresponds to a real value vector $\mathbf{K}'_{p,\mathrm{i}\cdot}\in\mathscr{R}^{n_o}$ in the p-th kernel space. Therefore, $s_p(\mathrm{o}_i,\mathrm{o}_j)=S_{p,ij}=\mathbf{K}'^{\top}_{p,\mathrm{i}\cdot}\mathbf{K}'_{p,\mathrm{j}\cdot}$ is a well-known linear kernel, which is positive semi-definite. Treating $s_p(\mathrm{o}_i,\mathrm{o}_j)$ as $k_1(\mathrm{o}_i,\mathrm{o}_j)$ in Lemma 7.1, since $\beta_p\geq 0$, consequently, $\beta_p s_p(\mathrm{o}_i,\mathrm{o}_j)$ is a positive semi-definite kernel according to Formula (1) of Lemma 7.1. Finally, as the wrapper kernel $s(\mathrm{o}_i,\mathrm{o}_j)=S_{ij}=\sum_{p=1}^{n_k}\beta_p S_{p,ij}$ is an accumulative summation of $\beta_p s_p(\mathrm{o}_i,\mathrm{o}_j)$, $s(\mathrm{o}_i,\mathrm{o}_\mathrm{j})$ is positive semi-definite by repeatedly adopting Formula (2) of Lemma 7.1 (treating $\sum_{p=1}^{q}\beta_p S_{p,ij}$ as $k_1(\mathrm{o}_i,\mathrm{o}_j)$, and treating $\beta_{q+1}S_{q+1,ij}$ as $k_2(\mathrm{o}_i,\mathrm{o}_j)$ for $1\leq q\leq n_k$). ■

Theorem 7.2 guarantees that the kernel matrix $\mathbf{S}$, constructed by kernel $s(\mathrm{o}_i,\mathrm{o}_j)$ as Equation (7.21), can be incorporated into the kernel k-means objective. This possibility is a fundamental property to support the effective unsupervised learning by UNTIE.

7.3.3 The Separability of UNTIE-Represented Data

The separability of a representation can be measured according to the overlap between object sets (e.g., clusters or classes) with repsect to the representation. The UNTIE-represented data has good separability since the resultant representation has the minimum normalized cut, which reflects the minimum overlap. Here, the normalized cut is firstly defined and then proved that the objective of UNTIE is equivalent to learning a representation with the minimum normalized cut.

Given a graph consisting of categorical data $G=<O,\mathbf{A}>$, where O is the set of categorical objects and where $\mathbf{A}$ is a non-negative and symmetric affinity matrix that contains the connected strength or similarity between objects, the *normalized cut* can be defined as following:

Definition 7.1. The *normalized cut* specifies the connection strength between two sets relative to the total connection strengths in a graph. Formally, the normalized cut between sets $O_1, O_2 \subseteq O$ is

$$normCut(O_1, O_2) = \frac{\sum_{i \in O_1, j \in O_2} \mathbf{A}(i,j)}{\sum_{i \in O_1, j \in O} \mathbf{A}(i,j)}.$$

Definition 7.1 shows that the normalized cut indicates the overlap between object sets. In other words, the *minimum normalized cut* reflects the maximum separability between clusters, which is essential for most of learning tasks (e.g., clustering and classification).

Theorem 7.3. *The objective of UNTIE in Equation* (7.22) *is equivalent to learning a representation with the minimum normalized cut.*

Proof. The objective of minimizing the normalized cuts between clusters can be formalized as,

$$\text{minimize} \quad \frac{1}{n_c} \sum_{j=1}^{n_c} normCut(O_j, O \setminus O_j),$$

where O_j refers to the set of objects in the j-th cluster and where $O \setminus O_j$ refers to the set of objects in the clusters instead of the j-th cluster. This objective function can be converted to a trace maximization problem according to Stella & Shi (2003) as follows:

$$\text{maximize} \quad \frac{1}{n_c} \text{Tr}(\mathbf{V}^\top \mathbf{A} \mathbf{V}),$$

where $\mathbf{V} = \mathbf{Z}(\mathbf{Z}^\top \mathbf{D} \mathbf{Z})^{-\frac{1}{2}}$; $\mathbf{Z} \in \{0,1\}^{n_0 \times n_c}$ is an indicator matrix for the cluster; and $\mathbf{D}$ is the diagonal matrix, in which (i,i)-entry is the sum of the i-th row in $\mathbf{A}$. If further relaxing the matrix $\mathbf{Z}$ to a real value matrix, and denoting $\mathbf{H} = \mathbf{D}^{\frac{1}{2}} \mathbf{V}$, the objective function can be converted to

$$\begin{aligned} \text{maximize} \quad & \frac{1}{n_c} \text{Tr}(\mathbf{H}^\top \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{H}) \\ \text{subject to} \quad & \mathbf{H} \in \mathscr{R}^{n_o \times n_c} \\ & \mathbf{H}^\top \mathbf{H} = \mathbf{I}_{n_c}. \end{aligned}$$

Let $\mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ be $\mathbf{S}$ in Equation (7.22). With the addition of a low rank regularization $\text{Tr}(\mathbf{S})$, this objective function becomes equivalent to the UNTIE objective function Equation (7.22). Therefore, the objective of UNTIE in Equation (7.22) is equivalent to learning a representation with the minimum normalized cut. ■

7.3.4 Convergence of the UNTIE Algorithm

Theorem 7.4. *The UNTIE algorithm converges to a local minimal solution in a finite number of iterations.*

Proof. Let y be the number of all possible partitions on a categorical data set C. Each partition can be represented by an indicator matrix $\mathbf{H}$. If two partitions differ, their indicator matrices also differ; otherwise, they are identical. Given the categorical data set C and the number of cluster n_c, y is finite. Therefore, there are a finite number of $\mathbf{H}$ on C. While applying the UNTIE algorithm to cluster C, a series of $\mathbf{H}$, i.e., $\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_{n_i}$, and a series of $\boldsymbol{\omega}$, i.e., $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \cdots, \boldsymbol{\omega}_{n_i}$, will be generated along the iterations. Given an indicator matrix $\mathbf{H}$ and a heterogeneity parameter $\boldsymbol{\omega}$, denote the loss value of UNTIE objective function Equation(7.22) as $l_{\mathbf{H},\boldsymbol{\omega}}$. Since kernel k-means and linear programming for Equation (7.24) to converge to minimal solutions, $l_{\mathbf{H},\boldsymbol{\omega}}$ is strictly decreasing (i.e., $l_{\mathbf{H}_1,\boldsymbol{\omega}_1} \geq l_{\mathbf{H}_2,\boldsymbol{\omega}_2} \geq \cdots \geq l_{\mathbf{H}_{n_i},\boldsymbol{\omega}_{n_i}}$). Assuming the number of iterations n_i is more than $y+1$, there are at least two same indicator matrices in the sequence (i.e., $\mathbf{H}_i = \mathbf{H}_j$, $1 \leq i \neq j \leq n_i$), where the corresponding optimized heterogeneity parameters are $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_j$, respectively. It is clear that $\boldsymbol{\omega}_i = \boldsymbol{\omega}_j$ since $\mathbf{H}_i = \mathbf{H}_j$. Therefore, $l_{\mathbf{H}_i,\boldsymbol{\omega}_i} = l_{\mathbf{H}_j,\boldsymbol{\omega}_i} = l_{\mathbf{H}_j,\boldsymbol{\omega}_j}$; that is, the value of the objective function does not change. If the value of the objective function is not changing, the UNTIE algorithm stops. Therefore, n_i is not more than $y+1$. Hence, the UNTIE algorithm converges to a local minimal solution in a finite number of iterations. ■

7.3.5 Computational Complexity of UNTIE

The time complexity of UNTIE is determined by two parts (i.e., building coupling spaces and learning heterogeneities). In building coupling spaces, the time cost depends on what kind of couplings UNTIE captures. In this chapter, UNTIE captures the intra- and inter-attribute couplings. Calculating intra-attribute coupling measures the frequency of each value, corresponding to complexity $O(n_v)$. Capturing inter-attribute couplings calculates the relationship between each value pair in each attribute pair, corresponding to the time cost $O(n_{mv}^2 n_a^2)$, where n_{mv} is the maximal number of values in attributes. Consequently, the entire time complexity of building the coupling spaces is $O(n_v + n_{mv}^2 n_a^2)$.

In heterogeneity learning, its time complexity is determined by the time cost of calculating eigenvectors and $\boldsymbol{\omega}$ and the number of iterations. If involving all data in each iteration, $O(n_o^3)$ is required to calculate eigenvectors, and $O((n_o^2)^{3.5} n_{\boldsymbol{\omega}}^2)$ is required to

solve linear optimization in calculating $\boldsymbol{\omega}$ (Zhu et al. 2018). Denoting the number of iterations as n_i, the time complexity of heterogeneity learning is $O(n_i n_o^3 + n_i (n_o^2)^{3.5} n_{\boldsymbol{\omega}}^2)$, where $n_{\boldsymbol{\omega}}$ refers to the number of elements in $\boldsymbol{\omega}$. If taking stochastic optimization, denoting the number of object pairs in each batch as n_b, the time complexity is only $O(n_b^3 n_i + n_b n_{\boldsymbol{\omega}} n_i)$. Since the batch size $n_b \ll n_o$ for a large data set, stochastic optimization is much more efficient if it can converge within a small number of iterations. Thus, stochastic optimization is recommended to solve the UNTIE objective; accordingly, the time complexity of UNTIE is $O(n_v + n_{mv}^2 n_a^2 + n_b^3 n_i + n_b n_{\boldsymbol{\omega}} n_i)$.

The space complexity of UNTIE with stochastic optimization is $O(n_o^2 n_{\boldsymbol{\omega}})$. For large categorical data, the space complexity is very high when full data optimization is used, which approaches $O(n_o^2)$. However, conducting stochastic optimization to obtain an approximate solution largely reduces the space complexity, since $n_b \ll n_o^2$. Therefore, stochastic optimization is adopted in UNTIE to tackle a large data set.

Overall, UNTIE has the time complexity $O(n_v + n_{mv}^2 n_a^2 + n_b^3 n_i + n_b n_{\boldsymbol{\omega}} n_i)$ and space complexity $O(n_b n_{\boldsymbol{\omega}})$. This result means UNTIE is scalable for large data. In addition, UNTIE can be further sped using parallel computing in both building coupling spaces and learning heterogeneities, which will be explored in the future work.

7.4 Experiments and Evaluation of UNTIE Performance

7.4.1 Parameter Settings of UNTIE

In the experiments, the UNTIE's default settings are as follows. The kernels used in UNTIE are 11 Gaussian kernels with a width from 2^{-5} to 2^5 and three Polynomial kernels with an order from 1 to 3. The stochastic optimization method Adam (Kingma & Ba 2014) is used to solve the UNTIE objective function, with an initial learning rate of 10^{-3}, a batch size of 20 and a maximum number of iterations of $1,000$. For the parameters of the baseline methods, the experiments take their recommended settings. UNTIE is implemented in Python 3.5 and Tensorflow r1.2; all experiments are conducted on a Windows 10 workstation with Intel i5-5300 U CPU@2.30GHz and 8GB memory.

7.4.2 Testing UNTIE Effectiveness

7.4.2.1 UNTIE-Enabled Clustering Performance

The UNTIE method is compared with (1) the categorical distance measure Hamming distance ("Hamming", for short); (2) five state-of-the-art categorical data representation methods including CDE (Jian et al. 2017), COS (Wang, Dong, Zhou, Cao & Chi 2015), DILCA (Ienco et al. 2012), Ahmad (Ahmad & Dey 2007), and Rough (Cao, Liang, Li, Bai & Dang 2012). The representations learned by vector-based representation methods, including UNTIE and CDE, are incorporated into k-means, which is probably the most popular clustering method and is sensitive to distance measurements; the representations learned by similarity-based representation methods, including COS, DILCA, Ahmda, Rough, and Hamming, are fed into k-modes, which represents the most commonly used clustering method for categorical data. To evaluate the performance of heterogeneity learning, UNTIE is compared with its variant which concatenates the representations in the coupling spaces without heterogeneity learning (denoting as Couplings), which is incorporated into k-means.

The comparison results are shown in Table 7.1. The best results are highlighted in bold, and Δ is the ratio of UNTIE's improvement over the best results of other measures. On half of the data sets, UNTIE performs significantly better than do the compared methods. For example, the F-score improves 51.72% on DNAN and 30.75% on Dmg compared to the best-performing methods DILCA and COS. On another half of the data sets, UNTIE achieves the same or results comparable to other methods. For example, the F-scores of UNTIE and CDE are both 56.56% on Mof and 54.8% on Tic. The UNTIE method effectively captures the intrinsic categorical data characteristics by revealing the value and attribute couplings and the heterogeneity within and between these couplings to induce the representation. These embedded characteristics guarantee the effectiveness of UNTIE representations, ensuring that the UNTIE-enabled clustering can generally achieve better results than others.

To statistically compare UNTIE's performance with the above categorical data representation methods, the Friedman test and Bonferroni–Dunn test (Demšar 2006) are used. The χ_F^2 of Friedman test is 33.68, associated with a p-value of $1.98e^{-5}$. This result indicates that the performance of all the compared methods is not equal. Further, the Bonferroni–Dunn test evaluates the CD between UNTIE and other methods and shows the CD at p-value < 0.1 is 1.69. As shown in Table 7.1, UNTIE achieves an overall AR of 2.7, which is better than other measures. For example, it is 0.84 better than that

Table 7.1: Clustering F-score (%) with Different Embedding Methods: The value-based representations are fed into k-means and the similarity-based representations are fed into k-modes to get the clustering results. The best results are highlighted in bold. Δ indicates the UNTIE's improvement over the best results of the other measures. The AR of a method over all data sets with significant difference from others with respect to the Bonferroni-Dunn test (p-value < 0.1) is labelled by *.

Data set	UNTIE	Couplings	CDE	COS	Ahmad	DILCA	Rough	Hamming	Δ
Zoo	**76.12**	74.85	75.04	72.1	71.34	71.34	62.79	73.27	1.44%
DNAP	**95.28**	92.45	61.61	49.24	49.92	85.85	63.2	52.68	10.98%
Hay	**54.17**	**54.17**	52.85	38.98	33.76	32.87	38.92	33.06	2.50%
Hep	70.40	**73.64**	69.82	46.29	66.72	65.13	59.21	59.21	0.83%
Aud	34.99	34.48	32.18	27.71	**35.38**	31.77	22.36	29.05	0%
Hsv	**90.51**	88.36	89.65	88.36	88.36	88.79	87.04	86.64	0.96%
Spc	55.04	55.04	52.55	36.26	34.93	34.76	**57.63**	35.94	0%
Mof	**56.65**	44.69	**56.65**	50.18	50.22	48.68	50.62	50.98	0%
SoyL	**69.29**	64.88	62.19	60.1	56.84	59.42	46.41	55.31	11.42%
Prim	24.62	24.87	23.43	19.81	23.65	21.76	22.38	**26.19**	0%
Dmg	**97.51**	72.78	73.1	74.58	72.87	72.61	57.99	66.6	30.75%
Tr	34.86	34.86	54.63	35.32	35.32	35.32	**65.19**	54.22	0%
Wcs	93.91	95.58	**96.2**	94.28	95.12	95.49	94.44	89.98	0%
Crx	**85.49**	52.65	52.65	36.99	52.65	79.29	63.47	79.29	7.82%
Br	93.27	94.75	95.2	93.56	94.89	**95.25**	94.37	93.27	0%
Ma	82.77	**82.89**	81.66	80.06	81.66	82.65	80.67	81.5	0.15%
Tic	54.8	**62.61**	54.8	51.88	50.87	52.97	50.19	53.59	0%
Flr	37.08	31.2	32.44	35.79	34.2	35.59	38.85	**39.22**	0%
Titn	33.72	29.77	33.72	29.77	33.72	33.72	**36.27**	33.72	0%
DNAN	**89.79**	67.7	51.14	41.91	46.68	59.18	43.28	41.44	51.72%
Spl	79.73	42.29	**87.12**	31.31	47.34	45.87	42.79	42.48	0%
Krv	51.09	51.09	51.03	46.72	**55.17**	**55.17**	53.73	53.86	0%
Ld	**69.5**	45.82	48.03	53.91	51.83	61.08	32.65	28.82	13.79%
Ms	82.69	82.76	82.83	**82.91**	82.86	82.39	78.18	82.29	0%
Cnt	**33.20**	31.14	31.91	27.23	32.88	33.14	30.34	31.43	0.18%
AR	**2.7***	4.1*	3.54*	5.9	4.54	4.42	5.42	5.38	0.84

of the best state-of-the-art method CDE (3.54), and 2.68 better than Hamming (5.38). Regarding the CD, UNTIE's performance is significantly better than all state-of-the-art methods except CDE. Although UNTIE and CDE do not significantly differ under the Bonferroni–Dunn test at p-value < 0.1, UNTIE captures the heterogeneity in couplings that cannot be learned by CDE. Therefore, the performance of UNTIE is better than CDE in most cases, especially on data sets with complex structures and heterogeneous distributions. For example, on DNAN, UNTIE achieves 89.79%, while CDE achieves only 51.14% in terms of F-score. All the comparison results are shown in Figure 7.2, which reveals UNTIE is significantly ($p < 0.1$) better than almost all the compared categorical representation methods.

The results also show that UNTIE and Couplings achieve the overall performance of 2.7 and 4.1 AR, respectively. This result shows that heterogeneity learning contributes to an additional 1.4 AR over the representation of Couplings. The UNTIE method does not consistently beat Couplings over all data sets, showing that not all data sets involve strong heterogeneity. For example, on Hep, Ma and Tic, the couplings-enabled representations show better results, while both UNTIE and Couplings do not make significant improvement over other methods, which also demonstrates that the clustering labels in these data sets are not sensitive to the captured couplings and heterogeneity. This lack of significance may indicate other unknown complexities in these data sets to be further explored.

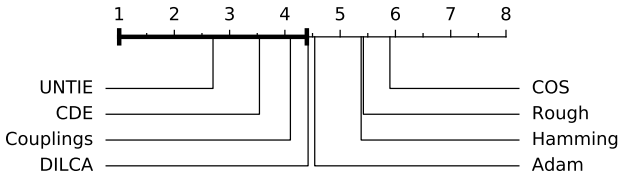


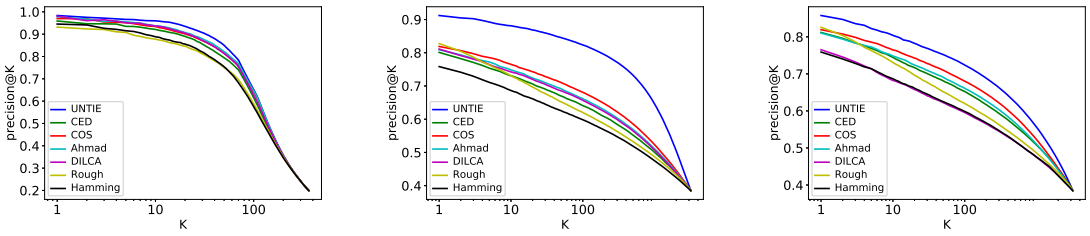
Figure 7.2: Comparison of UNTIE against the other representation methods per the Bonferroni–Dunn test. All representation methods with ranks outside the marked interval differ significantly (p < 0.1) from UNTIE.

7.4.2.2 UNTIE-Enabled Retrieval Performance

The experiment further tests the UNTIE representation performance of object retrieval, which is another task that depends heavily on data representation. Every object is used as a query, and its k-closest objects are retrieved per distance measure. The precision@k

(i.e., the fraction of the retrieved k objects that are the same-class neighbors) is reported. The experiment uses Euclidean distance for UNTIE- and CDE-represented data to compare with the distance measured by COS, DILCA, Ahmda, Rough and Hamming for retrieval. The three data sets Dmg, DNAN, and Spl are tested to evaluate the UNTIE-enabled retrieval performance.

Different from the clustering results, the precision@k of retrieval can demonstrate the quality of learned representation from local (when k is small) to global (when k is large). The results are shown in Figure 7.3, in which the precision of UNTIE-enabled retrieval consistently outperforms that of the other methods. Its performance reflects that UNTIE can capture more details of data distributions than can other representation methods, and this ability is enabled by learning hierarchical value-to-attribute couplings and heterogeneities.



(a) Precision on Dmg. (b) Precision on DNAN. (c) Precision on Spl.

Figure 7.3: The precision@k of different categorical data representation methods: A better metric yields a higher value.

7.4.3 Testing UNTIE Representation Quality

7.4.3.1 Reducing Inconsistency Caused by Heterogeneous Couplings

This section evaluates whether UNTIE can effectively reduce the inconsistency caused by learning heterogeneous couplings. First, it quantitatively measures the inconsistency by two indicators (i.e., intra- and inter-coupling heterogeneity indicators). Then, it analyzes the relations between UNTIE-enabled clustering results and the inconsistency indicated by these indicators.

The intra-coupling heterogeneity indicator (I_{intra}) measures the degree of heterogeneity in value distributions. It assigns a higher heterogeneity degree to couplings if each value in the couplings has more significantly diverse distributions. Intuitively, if a value has multiple distributions, its representations in different distributions

may be inconsistent with each other. The larger the differences its distributions have (i.e., higher heterogeneity degree), the stronger the inconsistency that may exist. In the experiment, I_{intra} compares the difference between distributions of a value per each ground-truth cluster, formalized as follows:

$$I_{intra} = \frac{\sum_{j=1}^{n_a} \sum_{i=1}^{n_v^{(j)}} \mathrm{NM}_{n_c} \left(\sqrt{\sum_{k=1}^{n_c} \left(\frac{|g^{(j)}(\mathsf{v}_i^{(j)}) \cap g^{(c)}(\mathsf{c}_k)|}{|g^{(j)}(\mathsf{v}_i^{(j)})|} \right)^2} \right)}{n_a}, \tag{7.25}$$

where c_k refers to the k-th cluster, n_c is the number of ground-truth clusters, function $g(\cdot)$ has the same definition as in Equation (7.1), and function NM_{n_c} normalizes a value to $[0,1]$ with respect to n_c clusters, which is defined as following:

$$\mathrm{NM}_{n_c}(x) = 1 - \frac{1-x}{1-\sqrt{\sum_{k=1}^{n_c} \left(\frac{1}{n_c}\right)^2}}. \tag{7.26}$$

The inconsistency within couplings caused by heterogeneous data distributions is indicated by I_{intra}, which will be large if each value has diverse distributions in each cluster or will be small if each value has similar distributions in each cluster. A larger I_{intra} indicates a stronger inconsistency within a coupling. In extreme cases, I_{intra} is 1 when each value appears only in a ground-truth cluster, and I_{intra} is 0 when every value has the same distribution in every ground-truth cluster.

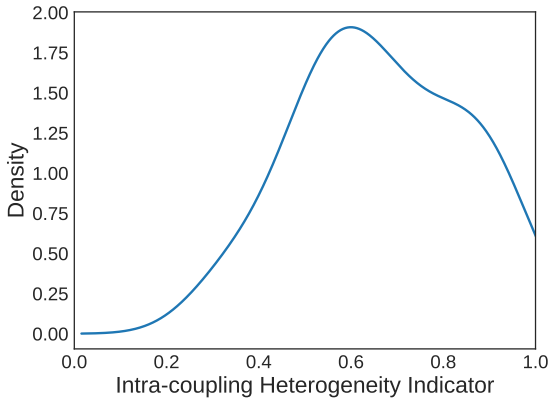


Figure 7.4: The probability density of intra-coupling heterogeneity indicator per kernel density estimation.

To show the distribution of intra-coupling inconsistency, this section illustrates the probability density of I_{intra} on 25 testing data sets in Figure 7.4 per kernel density estimation (KDE). The KDE method smoothly estimates the probability density of I_{intra}

within its range. A larger density indicates a higher probability of an inconsistency degree. As shown in Figure 7.4, most of data sets have strong inconsistency within couplings, which indicates the necessity of eliminating inconsistency in learning heterogeneous couplings.

The inter-heterogeneity indicator (I_{inter}) represents the degree of difference between couplings. Intuitively, the increase of differences between couplings (i.e., higher heterogeneity degree) may increase the inconsistency between categorical data representations. In the experiment, I_{inter} calculates the distance between coupling matrices to reflect the inter-coupling inconsistency. Here, a coupling matrix is a similarity matrix of the values in an attribute, and it is defined by a coupling learning method. For UNTIE, a coupling matrix is calculated by the Euclidean distance with respect to a coupling vector representation of values (e.g., the intra-attribute coupling value representation, as in Equation (7.1), or the inter-attribute coupling value representation, as in Equation (7.5)). For CDE, a coupling matrix is calculated by the Euclidean distance with respect to a value clustering representation. For COS and DILCA, the coupling matrices are their value similarity matrices before weighted integration. Denoting the k-th coupling matrix of the z-th attribute as $\mathbf{C}^{(z,k)}$, the inter-heterogeneity indicator is defined as follows:

$$I_{inter} = \sqrt{\frac{\sum_{k=1}^{n_m^{(z)}} \sum_{l=1}^{n_m^{(z)}} \frac{\sum_{i=1}^{n_v^{(z)}} \sum_{j=1}^{n_v^{(z)}} \left(\mathbf{C}_{ij}^{(z,k)} - \mathbf{C}_{ij}^{(z,l)}\right)^2}{n_v^{(z)2}}}{n_m^{(z)2}}}, \tag{7.27}$$

where $n_m^{(z)}$ is the number of couplings for the z-th attribute and where $\mathbf{C}_{ij}^{z,k}$ is the (i,j)-th entry of $\mathbf{C}_{ij}$. The value of I_{inter} will differ for divergent coupling learning methods. A larger I_{inter} indicates a stronger inconsistency between couplings, and I_{inter} will be large if each coupling matrix is substantially different from the others; it will be small if coupling matrices are similar. In an extreme case, I_{inter} is 0 when all coupling matrices are the same.

The probability density of I_{inter} on 25 testing data sets is shown in Figure 7.5 per KDE, which shows UNTIE and CDE involve a higher degree of inter-coupling heterogeneity compared to DILCA and COS. On one hand, the higher degree of inter-coupling heterogeneity provides richer information for representation. On the other hand, the higher degree of inter-coupling heterogeneity may be more likely to represent inconsistencies.

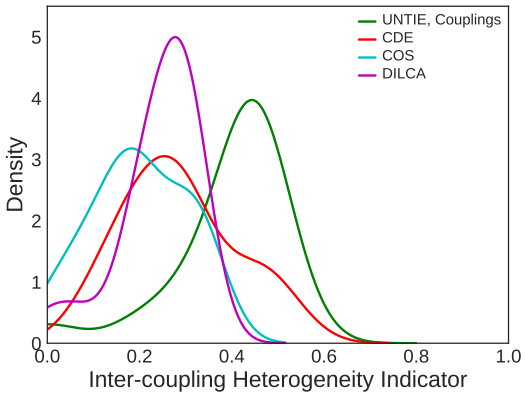


Figure 7.5: The probability density of inter-coupling heterogeneity indicator per kernel density estimation.

To evaluate whether UNTIE can effectively resolve the coupling inconsistency problem, this section analyzes the UNTIE-enabled clustering performance at different inconsistency levels indicated by the intra- and inter-coupling heterogeneity indicators. Considering the imbalanced distributions of these indicators as shown in Figure 7.4 and Figure 7.5, five inconsistency levels have been set, each of which contains the same number of data sets, according to the intra- and inter-coupling heterogeneity indicators. For example, the first level contains 20% data sets with the smallest values of I_{intra} or I_{inter}, while the fifth level contains 20% data sets with the largest values of I_{intra} or I_{inter}. The performance of a heterogeneous coupling learning method at an inconsistency level is calculated by averaging its enabled clustering rank in Table 7.1 on the data sets with the inconsistency level. The relations between the UNTIE-enabled clustering performance and the inconsistency level is illustrated in Figure 7.6 and Figure 7.7 per I_{intra} and I_{inter}, respectively.

As shown in Figure 7.6, UNTIE significantly outperforms its competitors on data sets at the inconsistency levels 2 to 5 in terms of I_{intra}. These levels have strong intra-coupling heterogeneity, as shown in Figure 7.4. While other methods ignore the intra-coupling heterogeneity, UNTIE captures it by learning value weights in kernel spaces with respect to Equation (7.10). As a result, UNTIE reduces the inconsistency and achieves better performance. In contrast, on the data sets with the inconsistency level 1, UNTIE does not show superiority. This lack of improvement is because these data sets do not have much coupling inconsistency, which further demonstrates that UNTIE gains better performance mainly from reducing the inconsistency between het-

erogeneous couplings.

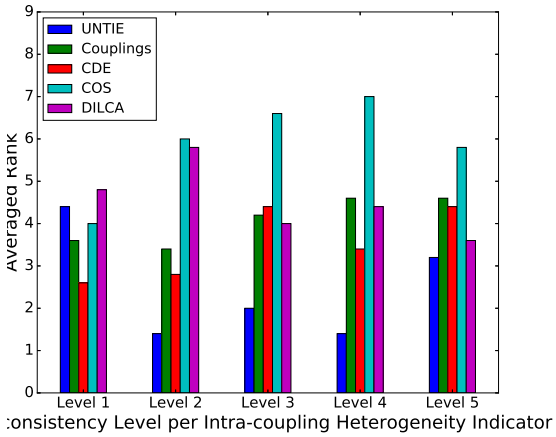


Figure 7.6: The UNTIE-enabled clustering performance on data sets with different inconsistency levels per the intra-heterogeneity indicator.

In Figure 7.7, the UNTIE-enabled clustering performance is much better than that enabled by existing heterogeneous coupling learning methods, especially on the data sets with higher inconsistency levels. Since the couplings represented by UNTIE may also incur inconsistency, as shown in Figure 7.5, UNTIE indeed reduces the inconsistency well to guarantee a good representation performance. This reduction is mainly contributed by the multiple kernels used in UNTIE, which enable this method to capture the fitness of couplings for different distributions. In Figure 7.7, UNTIE shows similar performance by simply combining multiple couplings on data sets with the inconsistency levels 2 and 3. This phenomenon indicates the heterogeneous couplings learned in the proposed method capture much richer data information compared to other methods and enable better performance when the inconsistency between these couplings is not so substantial. However, with the inconsistency increase, simply combining these couplings may worsen the results, as shown on the data sets with inconsistency levels 4 and 5 in Figure 7.7.

This experiment compares the UNTIE's similarity function $s(\cdot,\cdot)$ (defined in Equation (7.20)) with the similarity functions in the other similarity-based representation methods, by the (ϵ,γ)-good criterion. For the CDE-learned vector representation, the reversed Euclidean distance is used to measure the similarity between objects. In this experiment, the (ϵ,γ)-curves are drawn on four data sets (i.e., Mof, Dmg, Crx, and Br). The results are shown in Figure 7.8. It should be noted that only the ϵ that can guar-

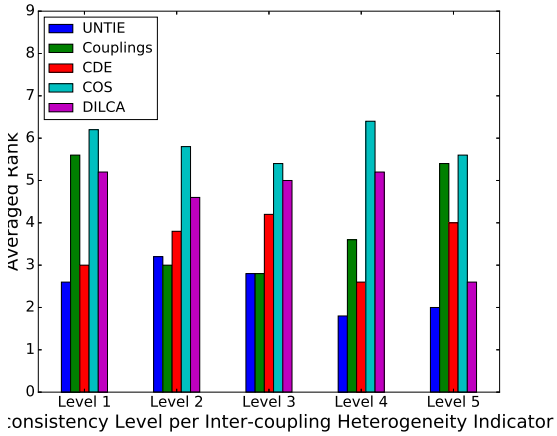


Figure 7.7: The UNTIE-enabled clustering performance on data sets with different inconsistency levels per the inter-heterogeneity indicator.

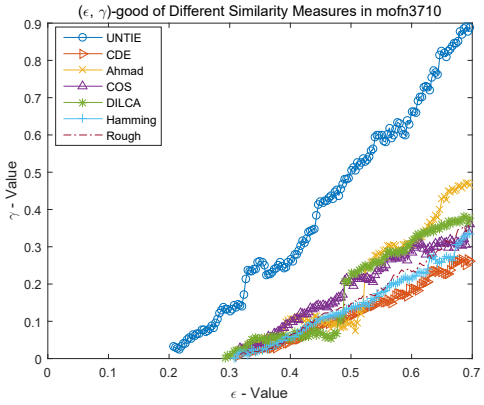
antee a non-negative margin (i.e., $\gamma \geqslant 0$) has been focused on. Therefore, Figure 7.8 displays only a part of the (ϵ, γ)-curve, in which $\gamma \geqslant 0$.

7.4.3.2 Goodness of the UNTIE-Enabled Metric

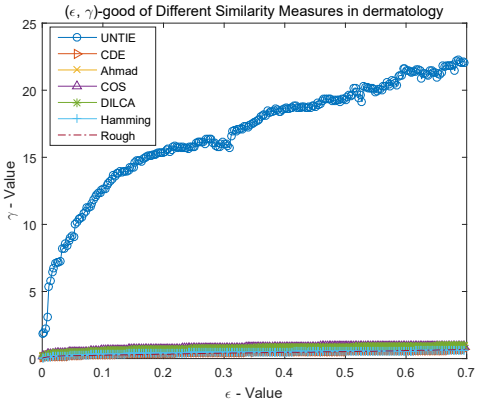
The results illustrate that UNTIE is better than its competitors in terms of the (ϵ, γ)-good criterion. The results also reveal the insight behind the clustering performance in Table 7.1. For data Dmg and Crx, the UNTIE-enabled clustering has much higher F-score than others, since UNTIE yields larger margins between different classes, as reflected by the (ϵ, γ)-good in Figures 7.8(b) and 7.8(c). For Mof, all methods obtain low F-score, and the UNTIE-enabled clustering achieves the same result as CDE, which is only slightly better than the other competitors. The reason is shown in Figure 7.8(a), where nearly 20% of the Mof data cannot be well separated by UNTIE (γ is 0 when ϵ is smaller than 0.2), while nearly 30% of that data cannot be well separated by its competitors. For other data sets (e.g., Br), all methods achieve good results since the (ϵ, γ)-good criterion indicates that these methods can separate the data sets well.

7.4.3.3 Visualization of UNTIE-Represented Data

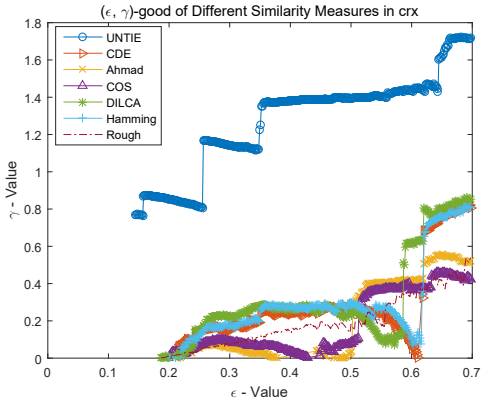
This section illustrates the visualization of UNTIE-represented data by converting it from high-dimensional representation to two-dimensional embedding by t-SNE (Maaten & Hinton 2008). For comparison, this section also visualizes the CDE-represented data and one-hot-represented data. Figure 7.9 shows the visualization of different rep-



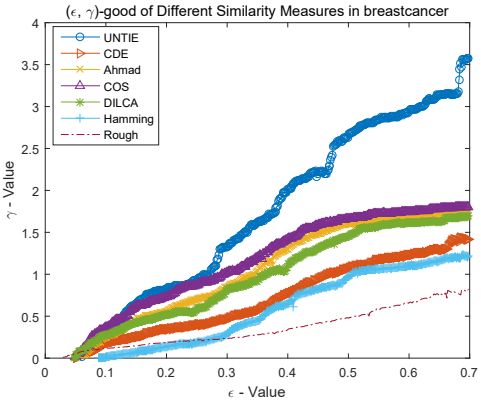
(a) (ϵ, γ)-curve on Mof.



(b) (ϵ, γ)-curve on Dmg.



(c) (ϵ, γ)-curve on Crx.



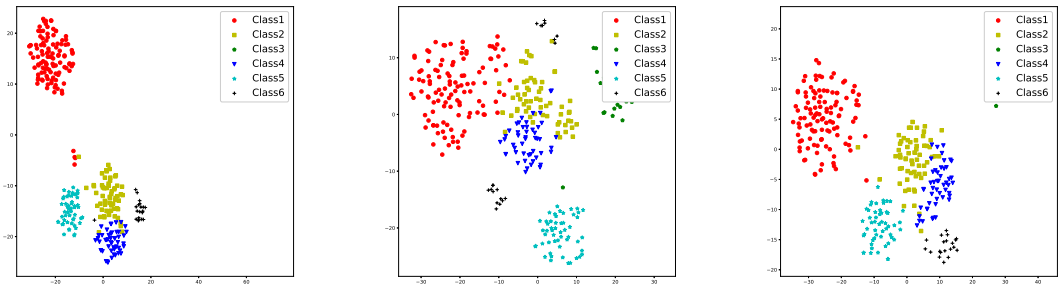
(d) (ϵ, γ)-curve on Br.

Figure 7.8: The (ϵ, γ)-curves of different transformed similarity measures: A better metric yields a better result.

resentation methods on Dmg. The UNTIE-represented data has a more compact distribution and leads to clearer boundaries between different clusters, as compared to other methods. It qualitatively demonstrates that UNTIE-represented data is more suitable for learning tasks, such as clustering and classification, because the proposed UNTIE learns the representation by optimizing the objective function Equation (7.22) (i.e., by minimizing the distance between objects within a cluster and maximizing the distance between objects in different clusters).

7.4.4 Testing UNTIE Efficiency

The efficiency of UNTIE is affected by the number of iterations to achieve convergence in Algorithm 3 and different data factors. This section first empirically evaluates the



(a) UNTIE-represented distribution.

(b) CDE-represented distribution.

(c) One-hot-represented distribution.

Figure 7.9: The visualization of different representation methods on Dmg. The UNTIE-represented data shows clearer boundaries between different clusters. The plotted two-dimensional embedding is converted from high-dimensional representation by t-SNE. Different symbols refer to different data clusters, per the ground truth.

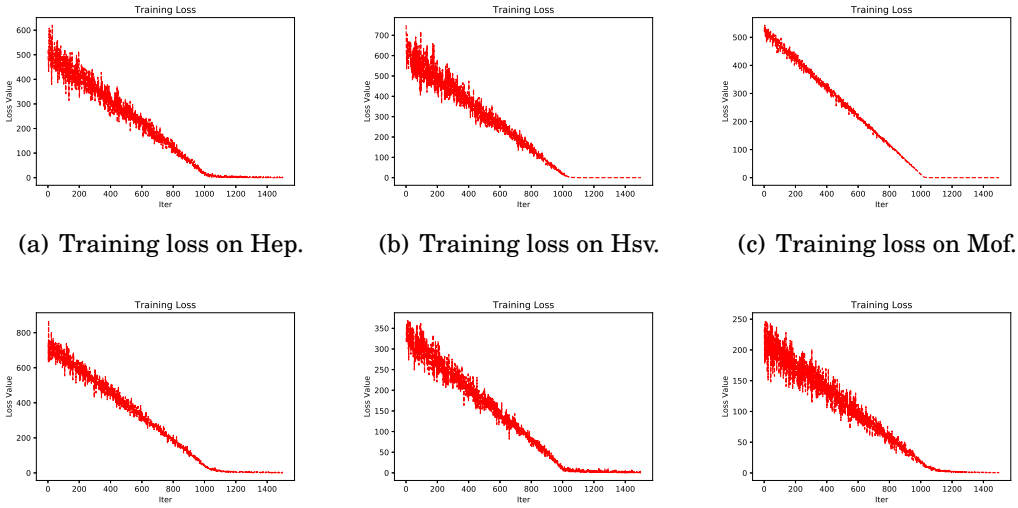
convergence speed of UNTIE, and then it evaluates the computational cost of UNTIE under different data factors.

7.4.4.1 Convergence of UNTIE

Six real data sets have been randomly selected to demonstrate the convergence of UNTIE. The optimization method for UNTIE is Adam, with the same setting as in Section 7.4.1. The training loss of objective function Equation (7.24) on these data sets is shown in Figure 7.10; the loss value converges rapidly at around 1,000 iterations. Since the batch size in each iteration is only 20, the time cost of 1,000 iterations is very low.

7.4.4.2 Computational Cost of UNTIE

Synthetic data are further generated to evaluate the computational cost of UNTIE in terms of the following data factors: the number of objects n_o, the number of attributes n_a, and the maximum number of values in each attribute n_{mv}. The default settings of these factors are as follows: n_o is 1,000, n_a is 10, and n_{mv} is 3. Three groups of data are generated and tune one of these factors for each group. For the first group of data, the number of objects is adjusted from 1,000 to 100,000. For the second group of data, the number of attributes is tuned from 10 to 100. For the third group of data, the maximum number of values in attributes is changed from 10 to 100. The time cost of UNTIE under each data factor is shown in Figure 7.11.



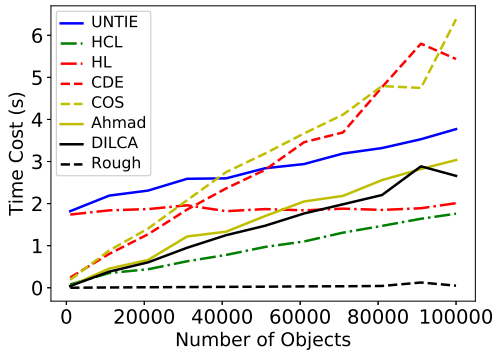
(a) Training loss on Hep. (b) Training loss on Hsv. (c) Training loss on Mof.

(d) Training loss on Dmg. (e) Training loss on Crx. (f) Training loss on Br.

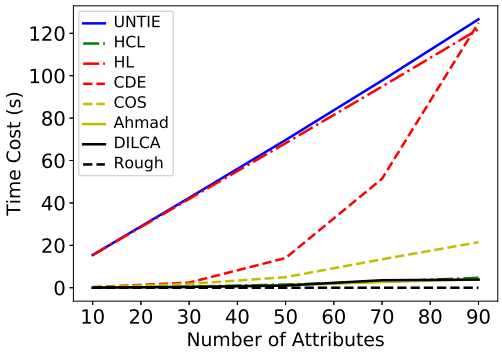
Figure 7.10: The UNTIE's training loss on different data sets. The stochastic optimization method for UNTIE is Adam (Kingma & Ba 2014), with an initial learning rate of 10^{-3} and a batch size of 20. The x-axis refers to the number of iterations, and the y-axis refers to the loss value of UNTIE's objective function Equation (7.24).

Figure 7.11(a) shows that the time cost of UNTIE is almost stable (from 1.8(s) to 3.6(s)), which demonstrates it has good scalability with respect to the amount of data n_o. The analysis shows that the minor time cost increase is caused by the built-in functions of Python when identifying categorical value location in the data. Since this cost increases with an extremely small proportion of data, it can be ignored when applying UNTIE. Figure 7.11(b) and Figure 7.11(c) demonstrate the time cost has an approximately linear relation with both n_a and n_{mv}, which is consistent with the time complexity of UNTIE analyzed in Section 7.3.5. These results also show that the main cost of UNTIE arises for heterogeneity learning (HL), which has a linear relation with both n_a and n_{mv}. Furthermore, the cost of building hierarchical coupling learning (HCL) has quadratic relation with n_a and n_{mv}. The reason for this cost is that UNTIE calculates the pairwise value relations when learning inter-attribute couplings. However, it only slightly affects the cost of UNTIE when n_a and n_{mv} are small. For categorical data with high dimensionality, a trade-off between sufficiently capturing couplings and preserving efficiency is required.

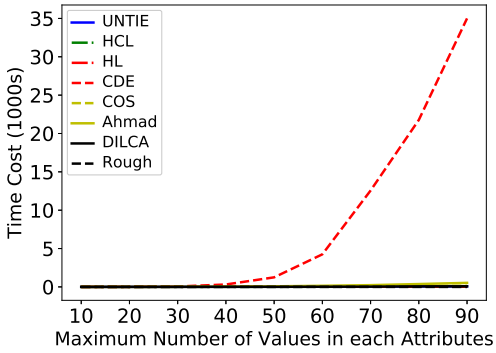
The computational cost of UNTIE and state-of-the-art methods are the same level in terms of data factor n_o. This data factor indicates all of these methods can handle large amount of data. The UNTIE method has a higher computational cost compared



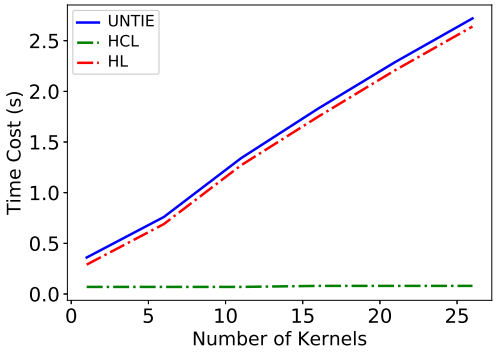
(a) Time cost w.r.t. number of objects.



(b) Time cost w.r.t. number of attributes.



(c) Time cost w.r.t. number of attribute values.



(d) Time cost w.r.t. number of kernels.

Figure 7.11: The UNTIE time cost with respect to data factors: object number n_o, attribute number n_a, and maximum number of attribute values n_{mv}. The solid line refers to the total time cost of UNTIE. The dotted line refers to the time cost of building the coupling spaces. The star line refers to the time cost of the heterogeneity learning.

to other methods in terms of n_a. As shown in 7.11(b), the higher cost is brought on by heterogeneity learning, which has a linear relation with n_a. For the hierarchical coupling learning, the cost of UNTIE is the same level as that of the state-of-the-art methods. With regard to the data factor n_{mv}, UNTIE is much more efficient than CDE, which is the state-of-the-art method with the best representation performance as shown in the previous experiments.

To evaluate the relation between time cost and the number of kernel functions, the number of kernels used in UNTIE is set from 1 to 30 and the computational cost of UNTIE is tested on the synthetic data set with the default data factors. The UNTIE time cost with a different number of kernels is shown in Figure 7.11(d). This figure shows that the UNTIE time cost is linear to the number of kernels with a very small

Table 7.2: KNN, SVM, RF and LR Classification F-score (%) with UNTIE and CDE. The best results are highlighted in bold typeface.

Data set	UNTIE-SVM	CDE-SVM	UNTIE-KNN	CDE-KNN	UNTIE-RF	CDE-RF	UNTIE-LR	CDE-LR
Zoo	**100**	88.00±18.33	**100**	**100**	**100**	**100**	**100**	**100**
DNAP	**94.42±6.81**	91.37±7.41	**87.19±10.79**	76.06±10.62	**89.32±8.81**	87.58±11.37	**94.41±6.03**	90.35±8.20
Hay	**82.23±6.22**	80.92±6.96	60.15±10.52	**62.38±10.33**	**82.48±9.00**	82.11±8.26	**82.08±7.29**	**82.08±7.29**
Lym	**87.06±12.02**	85.68±11.26	**82.22±11.08**	79.03±15.07	82.17±14.59	**84.16±11.31**	**83.12±12.90**	81.67±13.84
Hep	**48.13±8.65**	46.85±6.04	70.16±13.37	**70.47±14.52**	**66.61±11.12**	64.30±12.25	**70.39±16.85**	**70.39±16.85**
Aud	**73.41±7.29**	73.25±5.90	**48.19±9.47**	47.79±8.20	**63.80±10.24**	59.78±11.83	47.70±7.11	**66.29±11.26**
Hsv	96.71±3.88	**96.92±3.69**	**95.58±3.69**	92.54±4.18	**96.02±4.82**	95.13±4.61	93.74±5.00	**93.98±5.15**
Spc	67.60 ± 11.92	**68.46±10.88**	**55.41±9.95**	50.54±8.56	66.61±10.93	**67.13±12.08**	**69.48±12.52**	**69.48±12.52**
Mof	**100**	**100**	**87.18±6.29**	86.04±7.77	76.64±10.51	**78.31±12.16**	**100**	**100**
SoyL	90.94±3.83	**93.61±4.34**	93.70±4.26	**96.03±3.85**	**92.88±5.81**	92.69±6.12	**89.57 ±5.98**	88.57±6.97
AR	**1.35**	1.65	**1.35**	1.65	**1.35**	1.65	1.45	**1.55**

slope. Increasing the number of kernels only slightly affects the computational time of UNTIE. This slightly effect is consistent with the theoretical analysis, which indicates n_ω is linear to the time complexity of UNTIE. Here, n_ω has a linear relation with the number of kernels.

7.4.5 Testing UNTIE Flexibility

This section further demonstrates the flexibility of UNTIE by feeding it to four classifiers: k-nearest neighbor (KNN), support vector machine (SVM), random forest (RF), and logistic regression (LR). It randomly selects 90% of objects in each data set for training and uses the remainder for testing. To reduce the impact of noise and randomness, 20 sampling iterations generate 20 sets of training and test data for the experiments. The averaged classification performance and standard deviation are reported with respect to F-score (%). The vector representations learned by UNTIE and CDE are used as the input of these classifiers. The results comparing CDE-enabled classifiers are shown in Table 7.2 and illustrate that UNTIE can be adopted by different classifiers and enhances the classification performance on categorical data, as compared with the results of CDE-enabled classifiers.

7.4.6 Testing UNTIE Stability

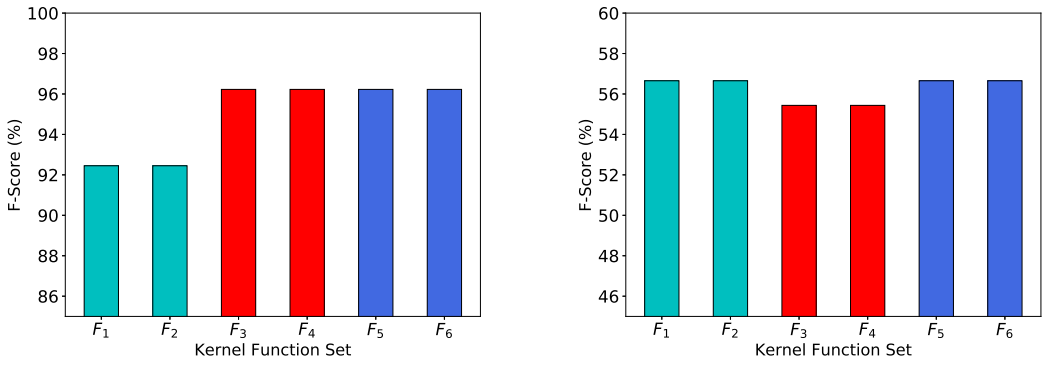
To evaluate the stability of UNTIE per the kernel functions in heterogeneity learning, three groups of kernel function sets with a varying number of functions are adopted. The first group contains only Gaussian kernels; the second group contains only polynomial kernels; and the third group mixes Gaussian and polynomial kernels. The kernel functions in each set are shown in Table 7.3. The clustering F-score enabled by UNTIE with respect to different kernel function sets on two data sets DNAP and Mof are illustrated in Figure7.12.

The UNTIE method is stable in terms of the number of kernel functions. As shown in Figure 7.12, UNTIE achieves the same clustering F-score with respect to kernel function sets in the same group, where kernel functions are of the same type but of different numbers. Although different kernel functions may generate different effects of UNTIE on different data sets (e.g., the Gaussian kernel family in group 1 enables better performance on DNAP, and the Polynomial kernel family in group 2 enables better performance on Mof), UNTIE can comprehensively learn information from multiple kernels while eliminating their redundancy and inconsistency. Accordingly, UNTIE always en-

ables the best clustering performance with the kernel function sets F_5 and F_6 in group 3, which involves kernel functions in both groups 1 and 2.

Table 7.3: Three Groups of Kernel Function Sets for UNTIE Stability Evaluation

Group	Set	Kernel Functions
Group 1	F_1	Gaussian kernels with width $\{2^{-3}, 2^{-2}, \cdots, 2^3\}$
	F_2	Gaussian kernels with width $\{2^{-5}, 2^{-4}, \cdots, 2^5\}$
Group 2	F_3	Polynomial Kernels with order $\{1, 2\}$
	F_4	Polynomial Kernels with order $\{1, 2, 3\}$
Group 3	F_5	Gaussian kernels with width $\{2^{-3}, 2^{-2}, \cdots, 2^3\}$ Polynomial Kernels with order $\{1, 2\}$
	F_6	Gaussian kernels with width $\{2^{-5}, 2^{-4}, \cdots, 2^5\}$ Polynomial Kernels with order $\{1, 2, 3\}$



(a) F-score (%) on DNAP.

(b) F-score (%) on Mof.

Figure 7.12: The clustering F-score (%) with UNTIE with respect to different kernel function sets: The same color indicates the same kernel function group.

7.5 Summary

This chapter studies unsupervised non-IID-completeness learning for categorical data representation, which embeds complicated coupling relationships and heterogeneities

hidden in complex categorical data. It proposes an unsupervised heterogeneous coupling learning method UNTIE. By modeling value-to-object hierarchical couplings and their complementary and inconsistent influence on representations, UNTIE reveals the non-linear relations between couplings and discloses the heterogeneous distributions within couplings. Both theoretical and empirical analyses show the effectiveness and efficiency of the proposed UNTIE for unsupervised categorical data representation.

The proposed HELIC (in Chapter 6) and UNTIE belong to non-IID-complete categorical data representation learning. They capture the static couplings and heterogeneities at some entity levels, such as the value-to-attribute-to-object level, but they do not consider the dynamic couplings and heterogeneities. The next chapter will propose an unsupervised coupling learning method on dynamic categorical data, based on the research outcome of the previous chapters, and it tackles the non-IID hard-learning problem.

Part IV

Non-IID-Hardness Learning

CHAPTER 8

UNSUPERVISED COUPLING LEARNING ON DYNAMIC CATEGORICAL DATA

8.1 Introduction

Dynamic categorical data representation is much harder than that of static categorical data because the couplings and heterogeneities embedded in data change over time. Consequently, effectively representing dynamic categorical data requires non-IID-hardness learning (see details in Section 2.5).

This chapter proposes an unsupervised hierarchical and heterogeneous coupling learning (UNICORN) method for unlabeled dynamic categorical data to achieve the objective of non-IID-hardness learningness. First, at the time point $t-1$, UNICORN learns the heterogeneous value-to-attribute-to-object hierarchical couplings with respect to each attribute. This step captures richer interactions and heterogeneities between attributes and between objects in complex categorical data than do existing methods based on the non-IID-completeness learning studied in Chapters 6 and 7.

Second, to sufficiently capture information in hierarchical heterogeneous couplings and fuse the heterogeneous couplings learned in unlabeled categorical data, UNICORN seamlessly wraps the above hierarchical heterogeneous coupling learning into a kernel learning by maximizing the informativeness of the wrapped kernel for an unsupervised representation. When more heterogeneous couplings are learned and integrated, this

unsupervised kernel informativeness maximization enables more information to be captured by the categorical representations built on these couplings (Xu et al. 2015, Zhu et al. 2018). As a result, the learned representations are more informative and suitable for various learning tasks than are existing approaches.

Lastly, at time t, to capture the dynamics of objects and their couplings, UNICORN re-learns heterogeneous couplings by involving the new observations available at this time point by hierarchically involving the data statistics of historical heterogeneous couplings learned at $t-1$ as the prior to regularize the current coupling learning objective. When the number of new observations at t is insufficient, the historical prior at $t-1$ will provide essential information of heterogeneous couplings within and between objects to the learner at time t. When heterogeneous couplings change from $t-1$ to t, integrating both new observations and the historical prior carries forward the transition, trend, and dynamics of the data and their interactions. This design prevents potential overfitting in dynamic categorical representation learning and improves the generalization ability of unsupervised representation learning.

Consequently, UNICORN is the first method to provide a generalized representation of hierarchical heterogeneous couplings from the value-to-attribute and object levels on unlabeled categorical data and to effectively and efficiently capture the variations of objects and their heterogeneous couplings in dynamic data. Comprehensive experiments on 12 diversified categorical data sets demonstrate that UNICORN can comprehensively capture hierarchical, heterogeneous, and dynamic couplings in dynamic categorical data, enable significantly better data representations, and result in substantial performance improvement for learning tasks such as clustering and retrieval, in comparison with three state-of-the-art similarity measures for categorical data.

8.2 The UNICORN Method

8.2.1 The UNICORN Architecture

The UNICORN architecture, working mechanism, and learning process are shown in Figure 8.1 for unsupervised hierarchical heterogeneous coupling learning on dynamic categorical data.

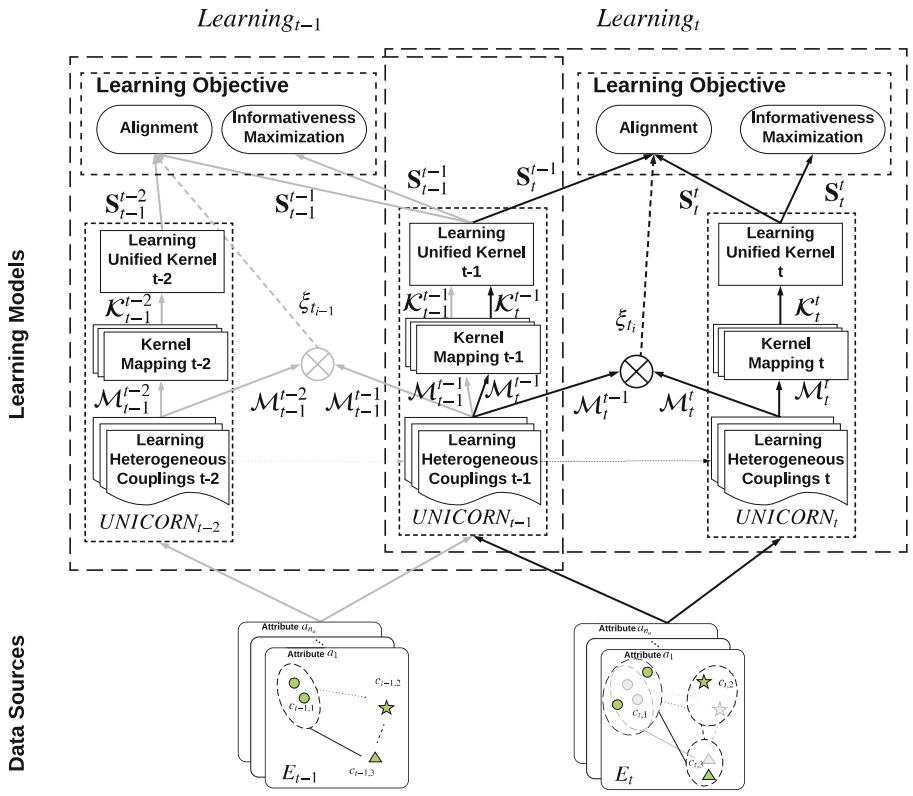


Figure 8.1: The UNICORN architecture: A multi-step dynamic learning process to capture and convert local heterogeneous couplings to unified global representations on dynamic categorical data.

8.2.1.1 UNICORN Working Mechanism

At each time point, as shown at each column in Figure 8.1, UNICORN undertakes a multi-step representation learning process. First, it reveals the heterogeneous (e.g., intra-attribute and inter-attribute) couplings embedded in the categorical data by inventing the corresponding coupling learning functions. These learned coupling functions form the corresponding base coupling spaces that capture the diverse low-level and observation-based relations. Second, these coupling spaces are further converted to multiple respective kernel spaces to build the kernelized coupling representations, where each coupling space can be mapped to one or multiple kernel spaces with respect to the corresponding kernel functions. Third, all these kernel spaces are further mapped to a unified kernel space, which presents a global representation of the entire data embedded with heterogeneous couplings. To obtain the optimal global representation, UNICORN maximizes the informativeness contained in the representation, which

essentially learns the optimal parameters for the UNICORN model.

The UNICORN method captures the dynamics of data across time points in terms of two strategies. First, the current data is fed to the UNICORN model learned at the previous time and is also used to relearn the UNICORN model. This relearning produces two unified global kernel representations as the outputs of the previous UNICORN and the new UNICORN models. Second, in learning the new UNICORN model, the various couplings captured by the previous UNICORN serve as priors to this new model (i.e., each type of couplings learned at the previous time is carried forward to the current time's coupling learning). The global kernel representation of the new UNICORN model is regularized to be aligned with the unified global kernel representation of the previous UNICORN model to obtain the coupling representation at the current time (e.g., as shown in Equations (8.2) and (8.7)).

8.2.1.2 Illustrating the UNICORN Learning Process

Below, the above UNICORN learning process is illustrated in terms of two time points, t and $t-1$. At time t, as shown at the right column of Figure 8.1, UNICORN learns a model $UNICORN_t$ in terms of multiple steps. First, as shown in Section 8.2.2, heterogeneous couplings hidden in the data E_t are learned with respect to different coupling learning functions (e.g., intra-attribute couplings $m^{(j)}_{Ia,t}(v^{(j)})$ as shown in Equation (8.1) and the inter-attribute couplings $m^{(j)}_{Ie,t}(v^{(j)})$ as shown in Equation (8.6) for each categorical value $v^{(j)}$ of attribute a_j). As a result, various coupling spaces are generated: for instance, the intra-attribute coupling spaces $\mathscr{M}^t_{Ia,t}$ (see Equation (8.2)), inter-attribute coupling spaces $\mathscr{M}^t_{Ie,t}$ (Equation (8.7)), and the aggregated entire attribute coupling spaces $\mathscr{M}^t_t$ (Equation (8.8)). Second, following the kernelization procedure described in Section 8.2.3, UNICORN transforms the learned coupling spaces to kernelized spaces $\mathscr{K}$. For example, Equation (8.9) illustrates how to construct a kernel space from a coupling space. Third, UNICORN further unifies all learned kernel spaces to a global kernel space in terms of the approach described in Section 8.2.4, as illustrated in Equation (8.15).

Consequently, as the result of undertaking the above UNICORN learning process, two UNICORN models are obtained (i.e., $UNICORN_{t-1}$ for $t-1$ and $UNICORN_t$ for t respectively), which consist of respective coupling spaces $\mathscr{M}^{t-1}_t$ and $\mathscr{M}^t_t$. These spaces further correspond to the unified kernel representations $\mathbf{S}^{t-1}_t$ and $\mathbf{S}^t_t$. To capture the data dynamics between times $t-1$ and t, UNICORN further involves the parameters learned for $UNICORN_{t-1}$ as priors of learning $UNICORN_t$. The representation $\mathbf{S}^t_t$ at t

on data E_t needs to be aligned with the representation $\mathbf{S}_t^{t-1}$ from $UNICORN_{t-1}$ on data E_t in terms of the degree ξ_t, which is calculated by comparing the difference between $\mathscr{M}_t^{t-1}$ and $\mathscr{M}_t^t$. Lastly, UNICORN maximizes the informativeness of $\mathbf{S}_t^t$ for time t, as shown in Equation (8.18).

8.2.2 Unsupervised Heterogeneous Dynamic Coupling Learning

The UNICORN method reveals and embeds diverse couplings in dynamic categorical data into respective coupling spaces in terms of designing corresponding coupling functions. This chapter simply incorporates the intra-attribute coupling function and inter-attribute coupling function designed by Zhu et al. (2018), who show the state-of-the-art results, while also focusing on how to learn such couplings in dynamic categorical data and construct their corresponding coupling spaces.

8.2.2.1 Representing Intra-Attribute Couplings

Intra-attribute couplings represent the interactions between values of an attribute, which can be described in terms of the value distributions in an attribute (Boriah et al. 2008, Wang, Chi, Zhou & Wong 2015). To capture intra-attribute couplings in dynamic data, the intra-attribute coupling function learns the intra-attribute couplings with the previously learned couplings as a prior. After learning the value couplings for an attribute at time $t-1$, UNICORN dynamically adjusts the learned value couplings with respect to the newly arrived categorical data observations at time t to obtain the value couplings at t.

Specifically, at time t, for a categorical value $v^{(j)}$ of the j-th attribute, the intra-attribute coupling function $m_{Ia,t}^{(j)}(\cdot)$ maps an intra-attribute coupling between this value and other categorical values in the j-th attribute to a one-dimensional intra-attribute coupling vector:

$$m_{Ia,t}^{(j)}(v^{(j)}) = \begin{cases} [\frac{|g_t^{(j)}(v^{(j)})|}{n_{o_t}}] & t=1 \\ \frac{m_{Ia,t-1}(j)\sum_{l=t_1}^{t-1} n_{o_l} + [|g_t^{(j)}(v^{(j)})|]}{\sum_{l=t_1}^{t-1} n_{o_l} + n_{o_t}} & t>1 \end{cases}, \tag{8.1}$$

where $g_t^{(j)}(\cdot)$ maps the value $v^{(j)}$ to a set of objects that have value $v^{(j)}$ in the j-th attribute for the categorical data E_t at time t.

Then, the intra-attribute coupling space $\mathcal{M}_{Ia,t}^{(j)}$ for attribute a_j at t is spanned by all the intra-attribute coupling vectors obtained for each value in the attribute per Equation (8.1), which is defined below:

$$\mathcal{M}_{Ia,t}^{(j)} = \{m_{Ia,t}^{(j)}(v^{(j)}) | v^{(j)} \in V_t^{(j)}\}. \tag{8.2}$$

Accordingly, for the categorical data E_t at time t with n_a attributes, the intra-attribute coupling space set $\mathcal{M}_{Ia,t}$ is obtained as follows: $\mathcal{M}_{Ia,t} = \{\mathcal{M}_{Ia,t}^{(1)}, \cdots, \mathcal{M}_{Ia,t}^{(n_a)}\}$.

8.2.2.2 Representing Inter-Attribute Couplings

The obtained intra-attribute coupling spaces can be viewed as a one-dimensional embedding of the categorical data space with respect to the couplings within each attribute, which do not consider the interactions between attributes. Accordingly, *inter-attribute couplings* refer to the interactions between attributes, which reflect the contextual (or semantic) information and relations of an attribute's values conditional on other attributes (Ienco et al. 2012). The inter-attribute couplings complement the within-attribute interactions with the between-attribute interactions, which together capture the hierarchical and heterogeneous couplings between objects.

Here, UNICORN represents the inter-attribute couplings with respect to the information conditional probabilities to reveal the distributions of an attribute value in the spaces spanned by the values of the other attributes. The UNICORN method extends the inter-attribute coupling function proposed by Zhu et al. (2018) to dynamic categorical data. Similar to dynamic intra-attribute coupling representation, the inter-attribute couplings at time t are adjusted with respect to both new observations at t and the inter-attribute couplings learned at $t-1$. At time t, for a value $v^{(j)}$ of attribute a_j and a value $v^{(k)}$ of attribute a_k, the information conditional probability function (ICPF) is calculated as follows:

$$p_t(v^{(j)}|v^{(k)}) = \begin{cases} p_t^1(v^{(j)}|v^{(k)}) & t = 1 \\ p_t^2(v^{(j)}|v^{(k)}) & t > 1 \end{cases}, \tag{8.3}$$

where

$$p_t^1(v^{(j)}|v^{(k)}) = \frac{|g_t^{(j)}(v^{(j)}) \cap g_t^{(k)}(v^{(k)})|}{|g_t^{(k)}(v^{(k)})|}, \tag{8.4}$$

$$p_t^2(v^{(j)}|v^{(k)}) = \frac{p_{t-1}(v^{(j)}|v^{(k)}) \sum_{l=t_1}^{t-1} |g_l^{(k)}(v^{(k)})| + |g_t^{(j)}(v^{(j)}) \cap g_t^{(k)}(v^{(k)})|}{\sum_{l=t_1}^{t-1} |g_l^{(k)}(v^{(k)})| + |g_t^{(k)}(v^{(k)})|}, \tag{8.5}$$

The operator $\cap$ returns the intersection of two sets. Based on the ICPF, the inter-attribute coupling learning function $m_{Ie,t}^{(j)}(v^{(j)})$ embeds interactions between value $v^{(j)}$ and other attributes as a $|V_{*,t}|$-dimensional inter-attribute coupling vector,

$$m_{Ie,t}^{(j)}(v^{(j)}) = \left[\begin{array}{ccc} p_t(v^{(j)}|v_{*1}), & \cdots, & p_t(v^{(j)}|v_{*|V_{*,t}|}) \end{array} \right]^\top, \tag{8.6}$$

where $V_{*,t} = \{V_t^{(k)} | k = 1, \cdots, n_a, k \neq j\}$ is a set of values that are in all attributes instead of a_j and where $v_{*\cdot} \in V_{*,t}$ is a categorical value in set $V_{*,t}$.

For the j-th attribute, UNICORN obtains its inter-attribute coupling space $\mathcal{M}_{Ie,t}^{(j)}$ that is spanned by the inter-attribute coupling vectors obtained for this attribute, as per Equation (8.6):

$$\mathcal{M}_{Ie,t}^{(j)} = \{m_{Ie,t}^{(j)}(v^{(j)}) | v^{(j)} \in V_t^{(j)}\}. \tag{8.7}$$

Accordingly, for categorical data E_t at time t with n_a attributes, UNICORN obtains all of its inter-attribute coupling spaces $\mathcal{M}_{Ie,t}$, $\mathcal{M}_{Ie,t} = \{\mathcal{M}_{Ie,t}^{(1)}, \cdots, \mathcal{M}_{Ie,t}^{(n_a)}\}$ for time t.

Each inter-attribute coupling learning function projects a categorical value into a higher-dimensional space if $|V_{*,t}| > 2|V_t^{(j)}| - 1$, since the dimensionality of all obtained inter-attribute coupling spaces equals $|V_{*,t}| - |V_t^{(j)}|$, while the degree of freedom (equivalent to dimensionality when transforming categorical values to dummy variables) of the j-th attribute is $|V_t^{(j)}| - 1$. In this way, the value couplings with respect to the influence of other attributes are captured, which collaborate with the intra-attribute couplings to form a complete representation of categorical attribute space with respect to both within- and between-attribute interactions.

8.2.3 Transforming Heterogeneous Couplings to Kernelized Spaces

With the intra-attribute coupling spaces $\mathcal{M}_{Ia,t}$ and the inter-attribute coupling spaces $\mathcal{M}_{Ie,t}$ learned at time t for data E_t, UNICORN further constructs the entire coupling spaces $\mathcal{M}_t$ for the data at t; that is,

$$\mathcal{M}_t = \mathcal{M}_{Ia,t} \cup \mathcal{M}_{Ie,t}, \tag{8.8}$$

where $\mathscr{M}_t$ consists of a collection of heterogeneous coupling spaces and where each space corresponds to a specific intra- or inter-attribute coupling.

To enable the heterogeneous couplings in the learned coupling space set to be mathematically alignable and tractable to each other while also retaining their own characteristics, UNICORN further adopts multiple kernels to transform each coupling space into its corresponding kernel spaces, where each kernel space corresponds to the transformed coupling space with respect to a particular kernel mapping function. Accordingly, UNICORN obtains a set of n_k ($= |\mathscr{M}_t| \times |F|$, where F is the set of kernel functions for the transformation) kernel spaces for time t: $\mathscr{K} = \{\mathscr{K}_{t,1}, \mathscr{K}_{t,2}, \cdots, \mathscr{K}_{t,n_k}\}$, where the p-th ($p \leq n_k$) space is spanned by a kernel matrix $\mathbf{K}_{t,p}$, which is constructed from a coupling space $\mathscr{M}_{t,j}$ with respect to a kernel function $k_p(\cdot,\cdot)$ for attribute a_j. Denoting $\mathsf{m}_{t,i}$ as a vector in $\mathscr{M}_{t,j}$ corresponding to the i-th categorical value, $\mathbf{K}_{t,p}$ can be represented as follows:

$$\mathbf{K}_{t,p} = \begin{bmatrix} k_p(\mathsf{m}_{t,1}, \mathsf{m}_{t,1}) & \cdots & k_p(\mathsf{m}_{t,1}, \mathsf{m}_{t,n_{v_t^*}}) \\ k_p(\mathsf{m}_{t,2}, \mathsf{m}_{t,1}) & \cdots & k_p(\mathsf{m}_{t,2}, \mathsf{m}_{t,n_{v_t^*}}) \\ \vdots & \ddots & \vdots \\ k_p(\mathsf{m}_{t,n_{v_t^*}}, \mathsf{m}_{t,1}) & \cdots & k_p(\mathsf{m}_{t,n_{v_t^*}}, \mathsf{m}_{t,n_{v_t^*}}) \end{bmatrix}, \tag{8.9}$$

where $n_{v_t^*}$ is the number of categorical values represented in the coupling space $\mathscr{M}_{t,j}$.

8.2.4 Building a Unified Global Representation

To reveal the heterogeneity within a coupling space, UNICORN learns the weights of categorical values in each corresponding kernel space. As the multiple kernel spaces mapped from one coupling space capture different value characteristics (e.g., distributions), the obtained kernelized spaces still share heterogeneities, which challenges the unification of all kernel representations. Therefore, UNICORN learns a set of transformation matrices $\{\mathbf{T}_{t,1}, \mathbf{T}_{t,2}, \cdots, \mathbf{T}_{t,n_k}\}$ to reconstruct another set of kernel spaces $\mathscr{K}'_t = \{\mathscr{K}'_{t,1}, \cdots, \mathscr{K}'_{t,n_k}\}$, in which the p-th kernel matrix $\mathbf{K}'_{t,p}$ reflects only the p-th kernel-sensitive distribution, capturing one aspect of the corresponding coupling.

The reconstructed kernel spaces are *coupling-specific heterogeneous kernel spaces*. Accordingly, each coupling is mapped to several kernels in $\mathscr{K}'_t$ to reflect the respective heterogeneities in the coupling. Kernel matrix $\mathbf{K}'_{t,p}$ is obtained as follows:

$$\mathbf{K}'_{t,p} = \mathbf{T}_{t,p} \cdot \mathbf{K}_{t,p}. \tag{8.10}$$

The UNICORN method regulates $\mathbf{T}_{t,p}$ as a diagonal matrix:

$$\mathbf{T}_{t,p} = \begin{bmatrix} \alpha_{t,p1} & 0 & \cdots & 0 \\ 0 & \alpha_{t,p2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{t,pn_v} \end{bmatrix}. \tag{8.11}$$

As a result, $\alpha_{t,pi}$ is the weight of the i-th categorical value (corresponding to the i-th row) in the p-th kernel space at time t, where the representation of the i-th row is

$$[k_p(\mathsf{m}_{t,i}, \mathsf{m}_{t,1}), \cdots, k_p(\mathsf{m}_{t,i}, \mathsf{m}_{t,n_{v_t^*}})].$$

The larger $\alpha_{t,pi}$ implies the i-th value has stronger coupling in the coupling space corresponding to the p-th kernel space.

To further capture the heterogeneity between couplings, UNICORN learns the contribution of each coupling-specific kernel space to the final representation. Specifically, it first defines a similarity measure between objects in the coupling-specific kernel space, and it then learns the weight of each kernel space based on this similarity measure to reflect the contribution of the coupling-specific kernel space. UNICORN uses $\mathbf{K}'_{t,p,\mathrm{i}\cdot}$—that is, the i-th row in $\mathbf{K}'_{t,p}$—to denote the object $o_{t,i}$ in the p-th heterogeneous kernel space. The similarity between the i-th and j-th objects in the kernel space $\mathbf{K}'_{t,p}$ with respect to a linear kernel is calculated below:

$$S_{t,p,ij} = \mathbf{K}'^{\top}_{t,p,\mathrm{i}\cdot} \mathbf{K}'_{t,p,\mathrm{j}\cdot}. \tag{8.12}$$

Per Equation (8.10), Equation (8.12) equals

$$S_{t,p,ij} = \mathbf{K}^{\top}_{t,p,\mathrm{i}\cdot} \mathbf{T}^{\top}_{t,p} \mathbf{T}_{t,p} \mathbf{K}_{t,p,\mathrm{j}\cdot}. \tag{8.13}$$

With all the coupling-specific kernel spaces $\{\mathcal{K}'_{t,1}, \cdots, \mathcal{K}'_{t,n_k}\}$, UNICORN calculates the similarities between the i-th and j-th objects. All the similarities between the i-th and j-th objects are then aggregated to form the unified similarity representation $S_{t,ij}$. For example, below, they are integrated in terms of a linear combination:

$$S_{t,ij} = \sum_{p=1}^{n_k} \beta_{t,p} S_{t,p,ij}, \tag{8.14}$$

where $\beta_{t,p} \geq 0$ is the weight for the p-th base similarity at t, which can be learned to filter redundant information and integrate complementary information between couplings. Denoting a diagonal matrix $\boldsymbol{\omega}_{t,p} = \beta_{t,p}\mathbf{T}_t^\top \mathbf{T}_t$, Equation (8.14) can be rewritten:

$$S_{t,ij} = \sum_{p=1}^{n_k} \mathbf{K}_{t,p,\mathrm{i}\cdot}^\top \boldsymbol{\omega}_{t,p} \mathbf{K}_{t,p,\mathrm{j}\cdot}. \tag{8.15}$$

In this way, UNICORN simultaneously learns $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_t$ by learning $\boldsymbol{\omega}_t$, which is a heterogeneity parameter. The optimized $\boldsymbol{\omega}_t$ will guide to integrate the heterogeneous couplings into the final similarity $S_{t,ij}$.

The UNICORN method further wraps the $\boldsymbol{\omega}_t$ to a wrapper kernel $s_t(\cdot,\cdot) : O \times O \to \mathscr{R}$, as defined below:

$$s_t(\mathrm{o}_{t,i}, \mathrm{o}_{t,j}) = S_{t,ij}. \tag{8.16}$$

The kernel $s_t(\cdot,\cdot)$ is proved as a positive semi-defined kernel (see details in Section 8.3.1). The kernel properties of $s(\cdot,\cdot)$ will be used to learn the heterogeneity parameter $\boldsymbol{\omega}_t$. Accordingly, UNICORN constructs a kernel matrix $\mathbf{S}_t$ with respect to the kernel $s_t(\cdot,\cdot)$ and a set of objects O_t observed at time t as follows:

$$\mathbf{S}_t = \begin{bmatrix} s_t(\mathrm{o}_{t,1}, \mathrm{o}_{t,1}) & s_t(\mathrm{o}_{t,1}, \mathrm{o}_{t,2}) & \cdots & s_t(\mathrm{o}_{t,1}, \mathrm{o}_{t,n_o}) \\ s_t(\mathrm{o}_{t,2}, \mathrm{o}_{t,1}) & s_t(\mathrm{o}_{t,2}, \mathrm{o}_{t,2}) & \cdots & s_t(\mathrm{o}_{t,2}, \mathrm{o}_{t,n_o}) \\ \vdots & \vdots & \ddots & \vdots \\ s_t(\mathrm{o}_{t,n_o}, \mathrm{o}_{t,1}) & s_t(\mathrm{o}_{t,n_o}, \mathrm{o}_{t,2}) & \cdots & s_t(\mathrm{o}_{t,n_o}, \mathrm{o}_{t,n_o}) \end{bmatrix}. \tag{8.17}$$

The UNICORN method uses $\mathbf{S}_t$ as the similarity representation of O_t at time t.

8.2.5 The Learning Objective Function of UNICORN

The UNICORN method regularizes the dynamic learning process by involving previously learned heterogeneous couplings into the coupling learning at the current time. At time t, UNICORN feeds the observed categorical data E_t to a new model $UNICORN_t$ to generate a similarity representation, $\mathbf{S}_t^t$, and also feeds E_t to the previously learned model $UNICORN_{t-1}$ to generate another similarity representation $\mathbf{S}_t^{t-1}$ for E_t.

To optimize the $UNICORN_t$ model, the informativeness measure with centered kernel alignment is adopted as the kernel similarity measure (Brockmeier et al. 2017) to quantify the informativeness in the similarity representation. Furthermore, to adjust $UNICORN_t$ in terms of $UNICORN_{t-1}$, the centered kernel alignment (Cortes

et al. 2012) between $\mathbf{S}_t^t$ and $\mathbf{S}_t^{t-1}$ is used for the regularization. Accordingly, the objective function of UNICORN is given as follows:

$$\begin{aligned}\underset{\boldsymbol{\omega}_t}{\text{minimize}} \quad & \frac{n_{o_t}(1-\bar{s_t})}{\|\mathbf{H}\mathbf{S}_t^t\mathbf{H}\|_F\sqrt{n_{o_t}-1}} \\ & -\delta\xi_t\frac{\text{Tr}(\mathbf{H}\mathbf{S}_t^t\mathbf{H}\mathbf{H}\mathbf{S}_t^{t-1}\mathbf{H})}{\|\mathbf{H}\mathbf{S}_t^t\mathbf{H}\|_F\|\mathbf{H}\mathbf{S}_t^{t-1}\mathbf{H}\|_F}+\sigma\|\boldsymbol{\omega}_t\|_1,\end{aligned} \tag{8.18}$$

where $\bar{s_t} = \frac{1}{n_{o_t}}\mathbf{1}^\top\mathbf{S}_t^t\mathbf{1}$, $\mathbf{H} = \mathbf{I} - \mathbf{1}\mathbf{1}^\top$, where $\mathbf{I}$ is the identity matrix (a square matrix in which all the elements of the principal diagonal are ones and all other elements are zeros), where δ and σ are trade-off hyper-parameters, and where ξ_t is a value ranging from $[0,1]$ that is determined by the change of value distributions. The value of ξ_t can be calculated as follows:

$$\xi_t = \begin{cases} 0 & t = 1 \\ \frac{1}{n_a n_{v_t}}\sum_{j=1}^{n_a}\sum_{v\in V_t^{(j)}}\frac{<m_t^t(v),m_t^{t-1}(v)>}{\|m_t^t(v)\|\|m_t^{t-1}(v)\|} & t \neq 1 \end{cases}. \tag{8.19}$$

In this chapter, the stochastic optimization method Adam (Kingma & Ba 2014) is used to obtain an approximate optimal solution for the above learning objective function.

When the heterogeneity parameter $\boldsymbol{\omega}_t$ is learned, the similarity representation of $\boldsymbol{E}_t$ is obtained as $\mathbf{S}_t$. The corresponding vector representation $\mathbf{x}_{t,i}$ of $o_{t,i}$ is obtained by

$$\begin{aligned}\mathbf{x}_{t,i} =&[\sqrt{\boldsymbol{\omega}_{t,1,11}}\mathbf{K}_{t,1,\text{i}1},\sqrt{\boldsymbol{\omega}_{t,1,22}}\mathbf{K}_{t,1,\text{i}2},\cdots, \\ & \sqrt{\boldsymbol{\omega}_{t,p,jj}}\mathbf{K}_{t,p,\text{i}j},\cdots,\sqrt{\boldsymbol{\omega}_{t,n_k,n_k^*n_k^*}}\mathbf{K}_{t,n_k,\text{i}n_k^*}]^\top\end{aligned} \tag{8.20}$$

where $\boldsymbol{\omega}_{t,p,jj}$ refers to the value of the (j,j)-th entry of $\boldsymbol{\omega}_{t,p}$ corresponding to the p-th kernel matrix $\mathbf{K}_{t,p}$, and where n_v^* refers to the number of values in the attribute corresponding to the n_k-th kernel matrix $\mathbf{K}_{t,n_k}$. Consequently, the learned representation $\mathbf{x}_{t,i}$ of object $o_{t,i}$ is a numerical approximation of the categorical data, which can then be fed to any vector-based learning method for further learning tasks.

This chapter summarizes the learning process of UNICORN for dynamic categorical data representation in Algorithm 4.

Algorithm 4 The UNICORN Algorithm for Dynamic Categorical Data Representation

Require: A set of categorical data sets $\{E_1, E_2, \cdots, E_{n_t}\}$, a set of kernel functions $F = \{f_1(\cdot,\cdot), f_2(\cdot,\cdot), \cdots, f_{n_{|F|}}(\cdot,\cdot)\}$.

Ensure: A set of similarity representations $\{\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_{n_t}\}$, a set of vector representations $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{n_t}\}$.

for $t = 1 : n_t$ **do**
  Mapping categorical data set E_t to coupling spaces $\mathscr{M}_{Ia,t}$ and $\mathscr{M}_{Ie,t}$ according to Equations (8.2) and (8.7).
  Constructing the entire coupling spaces $\mathscr{M}_t = \mathscr{M}_{Ia,t} \cup \mathscr{M}_{Ie,t}$ according to Equation (8.8).
  Mapping coupling spaces $\mathscr{M}_t$ to multiple kernel spaces $\{\mathbf{K}_1, \cdots, \mathbf{K}_{n_k}\}$ by using each kernel function in F to transform each coupling space in $\mathscr{M}_t$ according to Equation (8.9).
  Constructing the similarity $S_{t,ij}$ of objects in kernel spaces according to Equation (8.15).
  Constructing the wrapper kernel function $s_t(\cdot,\cdot)$ according to Equation (8.16).
  if $i > 1$ **then**
    Calculating the kernel matrix $\mathbf{S}_t^{t-1}$ according to Equation (8.17) based on O_t and $s_{t-1}(\cdot,\cdot)$ with respect to the heterogeneity parameter $\boldsymbol{\omega}_{t-1}$ learned at time point $t-1$.
  end if
  Constructing the kernel matrix $\mathbf{S}_t^t$ according to Equation (8.17) based on O_t and $s_t(\cdot,\cdot)$.
  Calculating ξ_t according to Equation (8.19).
  Optimizing $\boldsymbol{\omega}_t$ by solving Equation (8.18).
  Calculating similarity representation $\mathbf{S}_t$ according to Equation (8.17) with respect to $\boldsymbol{\omega}_t$.
  Calculating vector representation $\mathbf{X}_t$ according to Equation (8.20) with respect to $\boldsymbol{\omega}_t$.
end for
return $\{\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_{n_t}\}$, $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{n_t}\}$

8.3 Theoretical Analysis of UNICORN Properties

8.3.1 The Positive Semi-Definite Property of UNICORN Wrapper Kernel

In the objective function Equation (8.18), UNICORN quantifies the informativeness in the similarity representation and measures the consistency of the current and previously learned models based on kernel matrices $\mathbf{S}_t^t$ and $\mathbf{S}_t^{t-1}$, which are calculated by the kernel function $s_t(\cdot,\cdot)$, defined in Equation (8.16). To ensure the validation of the objective function Equation (8.18), $s_t(\cdot,\cdot)$ should be a positive semi-definite kernel.

Here, this chapter proves the kernel $s_t(\cdot,\cdot)$ is positive semi-definite. Before the proof, a fundamental lemma of kernel properties is introduced.

Lemma 8.1. *If $k_1(o_i,o_j)$ and $k_2(o_i,o_j)$ are positive semi-definite kernels in $O\times O$, and a constant $a>0$, then the following $k(o_i,o_j)$ functions are positive semi-definite kernels:*

(1) $k(o_i,o_j)=ak_1(o_i,o_j)$ and

(2) $k(o_i,o_j)=k_1(o_i,o_j)+k_2(o_i,o_j)$.

This lemma can be found in Section 4.1 of Steinwart & Christmann (2008).

Theorem 8.1. *The wrapper kernel $s_t(o_i,o_j)=S_{t,ij}$ defined in Equation* (8.16) *is a positive semi-definite kernel.*

Proof. Given coupling spaces $\mathscr{M}_t$ and a set of kernel functions F, the i-th object $o_{t,i}$ corresponds to a real value vector $\mathbf{K}'_{t,p,\mathrm{i}\cdot}\in\mathscr{R}^{n_{o_t}}$ in the p-th kernel space. Denoting $s_{t,p}(o_{t,i},o_{t,j})=S_{t,p,ij}=\mathbf{K}'^{\top}_{t,p,\mathrm{i}\cdot}\mathbf{K}'_{t,p,\mathrm{j}\cdot}$, $s_{t,p}(\cdot,\cdot)$ is a linear kernel, which is positive semi-definite. Treating $s_{t,p}(o_{t,i},o_{t,j})$ as $k_1(o_{t,i},o_{t,j})$ in Lemma 8.1, since $\beta_{t,p}\geq 0$, consequently, $\beta_{t,p}s_{t,p}(o_{t,i},o_{t,j})$ is a positive semi-definite kernel according to Formula (1) of Lemma 8.1. Finally, as the wrapper kernel $s_t(o_{t,i},o_{t,j})=S_{t,ij}=\sum_{p=1}^{n_k}\beta_{t,p}S_{t,p,ij}$ is an accumulated summation of $\beta_{t,p}s_{t,p}(o_{t,i},o_{t,j})$, $s_t(o_{t,i},o_{t,j})$ is positive semi-definite by repeatedly adopting Formula (2) of Lemma 8.1 (treating $\sum_{p=1}^{q}\beta_{t,p}S_{t,p,ij}$ as $k_1(o_{t,i},o_{t,j})$, and $\beta_{q+1}S_{t,q+1,ij}$ as $k_2(o_{t,i},o_{t,j})$ for $1\leq q\leq n_k$). ■

8.4 Experiments and Evaluation of UNICORN Performance

8.4.1 Dynamic Categorical Data Sets

In order the evaluate the dynamic categorical data representation, instead of on several representative data set in Table 3.2 that may contain more complicated relations for the purpose of evaluating coupling learning performance (Jian et al. 2017, Zhu et al. 2018), the experiments are further conducted on two data sets KDDCUP99 (KDD) and Letters (Ltr), which are from web-log and decision-making areas, respectively. Following the approach in Bai et al. (2016), the experiments randomly produce 10 data partitions of each data set. Each data partition contains 12 windows, each with a random cluster distribution. The data characteristics and the window size (n_w) for the partition are shown in Table 8.1.

8.4.2 Parameter Settings of UNICORN

In the experiments, the kernels used in UNICORN are 11 Gaussian kernels with a width from 2^{-5} to 2^5 and three polynomial kernels, with an order from 1 to 3, and σ is set as $\frac{1}{n_k}$, similarly to the setting used by Zhu et al. (2018). The default value of δ is set as 0.01. The objective function of UNICORN is optimized by Adam (Kingma & Ba 2014) with the initial learning rate as 10^{-3}, the batch size as 20, and the number of iterations as 250. For the parameters in the compared methods, the experiments take their recommended settings.

8.4.3 Testing Effectiveness of Learning Dynamic Categorical Data

The experiment first evaluates whether UNICORN can capture heterogeneous couplings to improve its enabled clustering on dynamic categorical data. It compares UNICORN-enabled k-means clustering with three categorical data stream clustering methods: clustering categorical data streams with drifting concepts (CCDS-DC) (Bai et al. 2016), clustering categorical time-evolving data (CCTED) (Cao et al. 2010), and clustering concept-drifting categorical data (DCD) (Chen et al. 2009).

Table 8.1 demonstrates the UNICORN-enabled k-means performance, compared to CCDS-DC, CCTED, and DCD for dynamic categorical data clustering. In most cases,

UNICORN performs significantly better than the compared methods. For example, the F-score improves 84.16% on DNANominal and 63.83% on Led24, compared with the best-performing methods DCD and CCDS-DC, respectively. On average, UNICORN enables k-means to beat others by 1.63 with respect to AR. The χ_F^2 of the Friedman test is 20.70, associated with a p-value of $1.22e^{-4}$. As the Friedman test result indicates significance differences between the performances of the compared methods, the experiment further conducts the Bonferroni–Dunn test, which shows that the performance of UNICORN-enabled k-means is significantly better than that of other methods, as shown in Figure 8.2.

The above results show that UNICORN can effectively capture and represent heterogeneous couplings in dynamic categorical data, leading to high quality representations. However, although the compared categorical stream clustering methods incorporate incremental learning mechanisms, they still cannot beat UNICORN, as they do not capture such hierarchical and heterogeneous couplings in data.

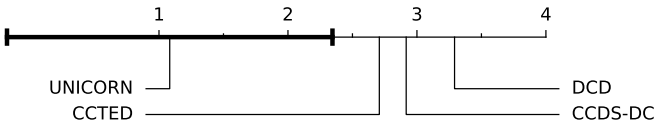


Figure 8.2: Comparison of UNICORN-enabled k-means clustering against categorical data stream clustering methods per the Bonferroni–Dunn test. All clustering methods with ranks outside the marked interval are significantly different ($p < 0.05$) from the UNICORN-enabled k-mean.

8.4.4 Comparison to State-of-the-Art Static Categorical Data Representation Methods

The experiments compare UNICORN with three state-of-the-art heterogeneous coupling learning methods: CDE (Jian et al. 2017), COS (Cao 2015), and DILCA (Ienco et al. 2012) for categorical data representation, as well as the deep neural network-based AE (Vincent et al. 2010), and the commonly used one-hot embedding representation, which does not consider the explicit couplings but implicit relations. Note that these baselines do not incorporate incremental learning, but rather learn the couplings based on observations in each data partition. The experiments compare them in terms of clustering and object retrieval.

Table 8.1: Clustering F-score of UNICORN-enabled K-Means vs. Different Categorical Data Stream Clustering Methods. The experiment shows the mean value of results on random partitions of a data set. The best results are highlighted in bold, and Δ shows the performance lift of UNICORN-enabled k-means over the best results of the baselines. The AR cross different data sets is reported to show the overall performance.

Data Information			**Clustering Performance**				**Lift**
Data	n_w	n_o	UNICORN	CCDS-DC	CCTED	DCD	Δ
Titn	125	2,201	**0.4724**	0.4609	0.4369	0.4283	2.49%
DNAN	250	3,186	**0.8932**	0.3451	0.4279	0.4850	84.16%
Spc	250	3,190	**0.8038**	0.3484	0.4399	0.4691	71.34%
Krv	250	3,196	0.5795	0.4119	**0.5977**	0.5637	0.00%
Ld	250	3,200	**0.6423**	0.3920	0.3099	0.3167	63.83%
Ms	250	5,644	**0.8707**	0.8551	0.8531	0.7789	1.82%
Ltr	125	20,000	**0.3412**	0.3175	0.2275	0.2275	7.45%
Krk	500	28,056	**0.2029**	0.1495	0.1676	0.1680	20.78%
Adt	500	30,162	**0.6112**	0.6030	0.6101	0.5426	0.18%
Cnt	500	67,557	**0.4653**	0.4099	0.4644	0.4059	0.20%
Cens	1,000	299,285	**0.6629**	0.6529	0.6520	0.5840	1.53%
KDD	1,000	494,021	**0.6616**	0.6208	0.6351	0.5732	4.18%
AR	-	-	**1.08**	2.92	2.71	3.29	1.63

The F-score results of UNICORN, CDE, COS, DILCA, AE, and one-hot-enabled clustering are reported in Table 8.2. The F-score of UNICORN-enabled k-means performs the best on 10 data sets, while achieving the best results on the other two data sets (having an F-score difference less than 0.01), and UNICORN achieves at most 27.69% improvement over the best-performing competitors. Overall, UNICORN improves 3.46 AR. The Friedman test shows large difference between the performance of UNICORN and that of the other methods (with χ^2_F 25.75 and a p-value of $2.47e^{-4}$). As shown in Figure 8.3, UNICORN achieves significantly ($p < 0.05$) better performance than its competitors do under the Bonferroni–Dunn test. Different from the state-of-the-art competitors, UNICORN not only reduces redundancy but also alleviates inconsistency in learning heterogeneous couplings. In addition, it inherits the historical coupling learning as

a prior to learn the current couplings, which enables a stronger generalization ability. Consequently, UNICORN significantly outperforms the three state-of-the-art coupling and relation representation methods.

Table 8.2: Clustering F-score of K-Means with Different Representations. The best results are highlighted in bold typeface, and Δ shows the UNICORN's improvement (lift) over the best results of the baselines. The AR across different data sets shows the overall performance.

	UNICORN Variants		**Representation Competitors**					**Lift**
Data	UNICORN	UNICORN-S	CDE	COS	DILCA	AE	One-hot	Δ
Titn	**0.4724**	0.4524	0.4418	0.4402	0.4597	0.4627	0.4564	2.10%
DNAN	**0.8932**	0.8756	0.7359	0.4252	0.6495	0.5836	0.6718	2.01%
Spc	0.8038	**0.8639**	0.7428	0.4281	0.6308	0.6009	0.6729	0.00%
Krv	0.5795	0.5328	0.5263	0.5756	**0.5832**	0.5478	0.5298	0.00%
Ld	**0.6423**	0.6271	0.4725	0.4877	0.5571	0.3127	0.3595	2.42%
Ms	**0.8707**	0.8577	0.8507	0.8345	0.8476	0.7401	0.8546	1.52%
Ltr	**0.3412**	0.2329	0.236	0.2672	0.217	0.2436	0.2404	27.69%
Krk	**0.2029**	0.1963	0.1994	0.1871	0.1998	0.1881	0.1918	1.55%
Adt	0.6112	0.6112	0.603	**0.6164**	0.6029	0.6054	0.603	0.00%
Cnt	**0.4653**	0.4425	0.4418	0.4579	0.4424	0.44309	0.4427	1.62%
Cens	**0.6629**	0.6529	0.6525	0.6528	0.6509	0.6572	0.6528	0.87
KDD	**0.6616**	0.64	0.6459	0.6346	0.6328	0.6097	0.6321	2.43
AR	**1.29**	3.38	4.71	4.54	4.67	4.75	4.67	3.46

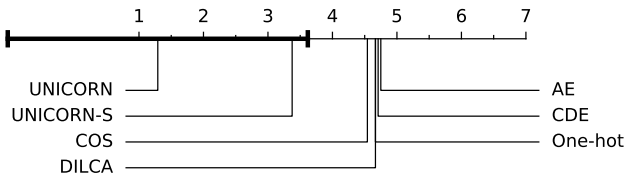
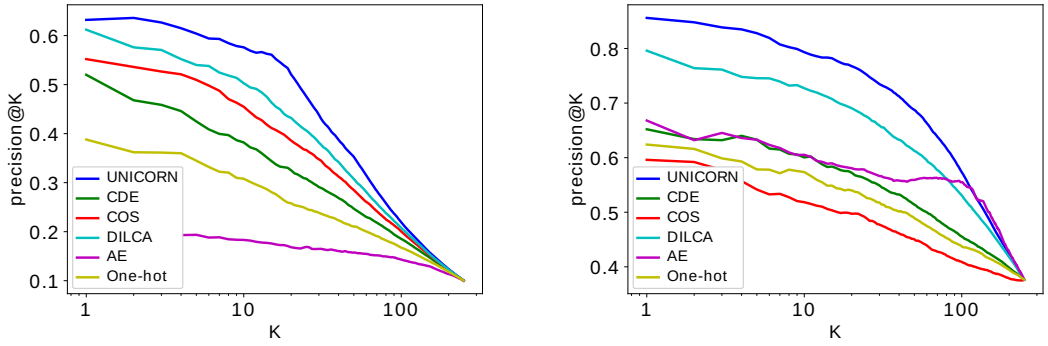


Figure 8.3: Comparison of UNICORN against other representation methods per the Bonferroni–Dunn test. All representation methods with ranks outside the marked interval differ significantly ($p < 0.05$) from UNICORN.



(a) Data set Ld. (b) Data set DNAN.

Figure 8.4: The precision@k-curve of different representations: A better representation yields a higher curve.

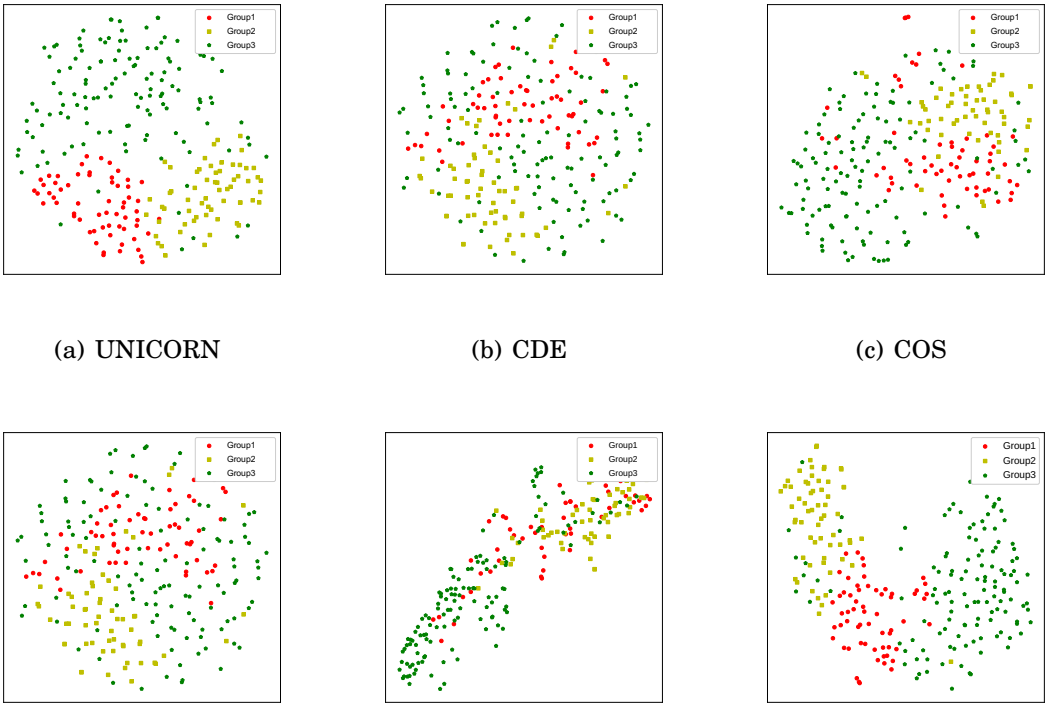
The object retrieval performance on the last partition window of two data sets D-NAN and Ld is further reported to demonstrate the representation performance of different methods. The results are shown in Figure 8.4, and the precision of UNICORN-enabled retrieval consistently outperforms the competitors. It reflects that UNICORN represents more information carried from the historical statistic prior. In contrast, other methods only learn from the observations in a specific data partition.

To further illustrate the value of representing hierarchical heterogeneous couplings in dynamic categorical data, this thesis visualizes the representation results generated by different methods. Figure 8.5 displays the outcomes with respect to a two-dimensional space by t-SNE (Maaten & Hinton 2008) converted from the high-dimensional representations made by each method on the DNAN data set. In comparison, objects belonging to the same group are more likely co-located with others. It also quantitatively demonstrates that the UNICORN-represented results are more suitable for general learning tasks.

8.4.4.1 Effectiveness in Learning Coupling Dynamics

The effectiveness of learning coupling dynamics over time is evaluated by comparing UNICORN with other heterogeneous coupling learning methods in clustering. The ARs over every four data partitions of all data sets are calculated to show the overall performance of these methods.

As shown in Figure 8.6, UNICORN achieves the best performance with the lowest ARs at every iteration. Its ranking follows a non-increasing trend, while its competitors



(a) UNICORN (b) CDE (c) COS

(d) DILCA (e) AE (f) One-hot

Figure 8.5: The Visualization of different representation methods: A better representation yields closer grouping results.

face more randomly distributed ranks over time. This result indicates UNICORN can effectively capture the dynamic couplings in the dynamic categorical data.

8.4.5 Ablation Study of UNICORN Design

The experiment further evaluates the contribution of each learning objective in UNICORN: learning heterogeneous coupling learning (enabled by informativeness maximization) and learning dynamic couplings (enabled by kernel alignment). To test the heterogeneous coupling learning performance, the experiment ablates the centralized kernel alignment in the UNICORN objective function to obtain a UNICORN variant UNICORN-S. The dynamic coupling learning performance is evaluated by comparing the difference between UNICORN and UNICORN-S.

The evaluation of UNICORN-S is shown in Table 8.2 and Figure 8.3. The results show that UNICORN-S outperforms the state-of-the-art competitors on most of da-

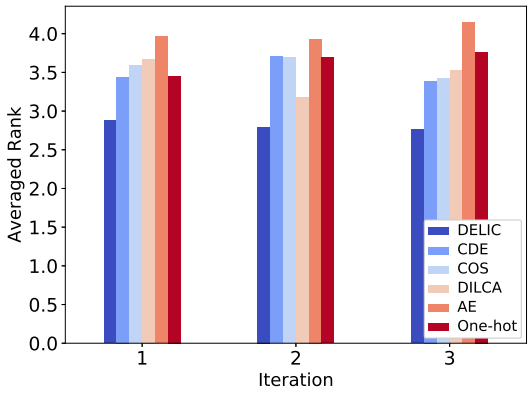


Figure 8.6: Averaged ranks over three iterations of data dynamics: Each iteration averages the ARs of four data partitions of all data sets. A better method incurs a lower AR in each iteration. The methods that can capture dynamic couplings will generate non-increasing ARs over iterations.

ta sets. Overall, it also gains 1.17 AR improvement over the 4.25 rank of the best-performing competitor, COS. This rank demonstrates that UNICORN captures richer heterogeneous couplings than the compared coupling learning methods.

Compared with UNICORN-S, UNICORN generally achieves the best performance on 11 data sets, with comparable performance on one remaining data set (while UNICORN still significantly outperforms other methods). This better performance is likely contributed by the historical statistics captured by UNICORN to enhance the heterogeneous coupling learning at the current time.

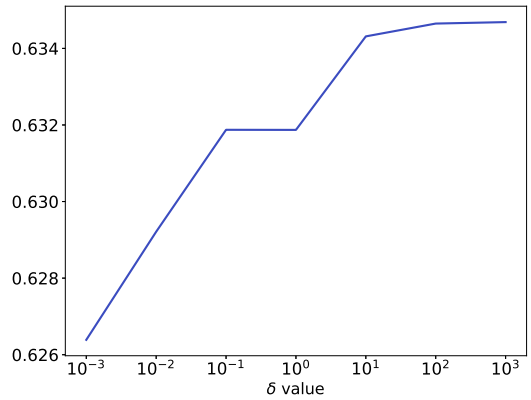


Figure 8.7: The UNICORN-enabled clustering F-score with respect to hyper-parameter δ.

8.4.6 Testing UNICORN Stability

The experiment evaluates UNICORN stability with regard to its hyper-parameter δ, which controls the impact of kernel alignment (the second term in the objective function Equation (8.18)). The larger δ is, the more historical information is carried forward to the current learning process. Figure 8.7 shows the UNICORN-enabled clustering F-score under different settings of δ on the Led24 data set. With the value of δ increasing, the F-score increases from 0.626 to 0.635. This small increase range indicates UNICORN is stable with respect to δ. Meanwhile, the F-score increase trend also demonstrates the importance of carrying forward historical heterogeneous couplings.

8.5 Summary

This chapter introduces a non-IID-hardness learning method (i.e., UNICORN) for dynamic categorical data representation. The UNICORN method learns the heterogeneous couplings and heterogeneities in categorical data based on the non-IID-completeness learning studied in Chapters 6 and 7, and it embeds these couplings and heterogeneities through an informativeness maximization objective. This method can incrementally capture the coupling and heterogeneity dynamics caused by new observations and efficiently integrate these dynamics with historical couplings and heterogeneities. These abilities enable non-IID-hardness learning for complex dynamic data representation. As a result, UNICORN generates superior representations for stable and dynamic categorical data enhancing various analytics tasks, as demonstrated in the experiments such as for clustering and objective retrieval. Comprehensive experiments demonstrate that it can effectively capture dynamic heterogeneous couplings, and its enabled representation quality is significantly better than three state-of-the-art dynamic categorical data representation methods in terms of their enabled clustering and object retrieval.

CHAPTER 9

CONCLUSIONS AND FUTURE DIRECTIONS

9.1 Conclusions

This thesis builds a theory of non-IID representation learning on complex categorical data, which consists of the following components: (1) coupling learning (studied in Chapter 4); (2) heterogeneity learning (studied in Chapter 5); (3) non-IID-complete learning (studied in Chapters 6 and 7); and non-IID-hard learning (studied in Chapter 8). This theory effectively captures and embeds comprehensive non-IIDness, from couplings, heterogeneities, heterogeneous couplings, and non-IID-completeness to non-IID-hardness, for complex categorical data representation. It also provides a serial of theoretical tools to analyze the effectiveness of non-IID representation learning.

9.1.1 Coupling Learning

The coupling learning component studies and models multiple hierarchical couplings in complex categorical data from different aspects. Specifically, it investigates intra-attribute couplings, inter-attribute couplings, and attribute-label couplings to form hierarchical and horizontal value-to-attribute-to-object couplings. It further proposes a non-IID similarity metric learning framework as well as its instantiation method to embed and integrate these couplings into a similarity space where the categorical data representation satisfies similarity metric properties. This component serves as one of

the key foundations (another is heterogeneity learning) of non-IID-complete learning and non-IID-hard learning.

9.1.2 Heterogeneity Learning

The heterogeneity learning component recognizes and captures the heterogeneities of multiple distributions and multiple dependencies in complex categorical data. To learn heterogeneous distributions, it provides a decoupled non-IID learning framework, which disentangles and models independent distributions that are mixed within each attribute. To learn heterogeneous dependencies, it introduces a kernel-based method to model the relations between these dependencies. It further proposes a nonparametric Bayesian embedding model to embed heterogeneities into categorical data representation, built on a Dirichlet multivariate multinomial mixture model. This component, together with the coupling learning component, builds the bases of non-IID learning theory in this thesis.

9.1.3 Non-IID-Complete Learning

The non-IID-complete learning component studies how to learn and integrate heterogeneous dependencies and distributions with complementarity and inconsistency. Based on the research of coupling learning and heterogeneity learning, it reports on both supervised and unsupervised non-IID-complete learning methods. These non-IID-complete learning methods jointly consider the redundancy, complementarity, and inconsistency in heterogeneous and hierarchical couplings. Based on these non-IID-complete learning methods, this component provides representation methods to comprehensively embeds couplings and heterogeneities within and between complex categorical data into a vector or similarity space. Compared with the existing categorical data representation methods, the representation methods provided by this component are much more effective when heterogeneous dependencies and distributions both exit in data (i.e., data is with the non-IID-completeness). This non-IID-complete learning component further serves as a foundation of the non-IID-hard learning, which learns non-IIDness at all entity levels.

9.1.4 Non-IID-Hard Learning

The non-IID-hard learning component studies how to learn heterogeneous couplings and heterogeneities at all entity level, where dynamics should be considered. At the first

time point, it learns couplings and heterogeneities by the non-IID-complete component. To handle dynamics, it incrementally updates the learned couplings and heterogeneities by learning from new observations with the learned heterogeneous non-IIDness as a prior. Further, it provides a non-IID-hard representation learning method for dynamic categorical data. To enable general-purpose downstream tasks, this method embeds the learned non-IIDness into a representation by maximizing the informativeness of this representation. Compared with the existing dynamic categorical data representation methods, the representation method provided by this component captures complex dynamic non-IIDness and thus achieves significantly better representation performance.

9.2 Future Directions

The research in this thesis can be further extended in several directions: (1) exploiting more efficient yet powerful non-IID representation methods for categorical data; (2) studying non-IID representation on more types of data; and (3) quantifying the non-IID data complexities.

9.2.1 Exploiting Other Non-IID Representation Methods for Categorical Data

This thesis builds several non-IID representation methods on complex categorical data with different kinds of non-IIDness. However, these methods have certain limitations. For example, they require designated coupling measurements, each of which captures only a type of coupling from a single perspective, and most of them are frequency-based methods. As a result, they cannot leverage comprehensive couplings for complex categorical data representation. To tackle this problem, a promising direction is to exploit how to adaptively learn implicit couplings that are complemented with the explicit couplings learned by designated measurements. Another limitation of the proposed methods is that they are highly time-consuming. All the proposed methods, instead of non-BEND, are kernel-based and consider pair-wised values and attributes relations. Consequently, they have high time complexity in terms of the number of values and attributes analyzed, and they may fail to represent categorical data with ultra-high dimensions and a large number of unique categorical values. How to efficiently represent categorical data yet effectively capture hierarchical and heterogeneous couplings and heterogeneities remains an open problem.

9.2.2 Studying Non-IID Representation on More Types of Data

This thesis studies non-IID representation only on categorical data. However, it builds a deep understanding of non-IIDness and provides several theoretical tools for non-IID representation learning that can be further extended to other types of data, such as numerical data, textual data, image data, spatio-temporal data, and their mixture with categorical data. Considering textual data as an example, heterogeneous couplings and heterogeneities exist between different words that determine various meanings of the same words and phrases, and these word-level couplings and heterogeneities further couple with each other to form sentence-level couplings and heterogeneities. Capturing and embedding these hierarchical and heterogeneous couplings and heterogeneities will enhance textual data analysis performance because they essentially determine the semantic meaning of textual data. Furthermore, it is also valuable to study the relations between non-IIDness in different types of data, which may provide a foundation to non-IID representation learning on heterogeneous data with different types.

9.2.3 Quantifying the Non-IID Data Complexities

This thesis proposes a series of methods to represent complex categorical data. However, it does not provide enough tools to quantify the non-IID data complexities. More tools are required to detect and quantify the couplings, heterogeneities, heterogeneous dependencies, heterogeneous distributions, non-IID complete, and non-IID hard in complex categorical data. Quantifying these complexities can provide a valuable guideline and theoretical foundation for choosing and designing non-IID representation methods catering to specific data characteristics.

APPENDIX

A.1 List of Notations for Static Data

Φ	categorical data distributions
E	categorical data tuple
O	object set
A	attribute set
V	categorical value set of all attributes
$V^{(j)}$	categorical value set of the j-th attribute
C	class or cluster set
o_i	the i-th object in O
a_i	the i-th attribute in A
v_i	the i-th categorical value in V
$v_i^{(j)}$	the categorical value of o_i in a_j
$\mathsf{v}_i^{(j)}$	the i-th categorical value in $V^{(j)}$
m_i	the vector corresponding to i-th value in a coupling space
c_i	the i-th class (or cluster)

$\boldsymbol{\omega}$	the heterogeneity parameter
n_o	the number of objects in O
n_a	the number of attributes in A
n_v	the number of categorical values in V
$n_v^{(j)}$	the number of categorical values in $V^{(j)}$
n_k	the number of kernel matrices transformed from coupling spaces
n_c	the number of classes (or clusters)
n_{oc}	the size of c-th class (or cluster)
n_{mv}	the maximal number of values in attributes
n_{av}	the average number of attribute values
$n_{\boldsymbol{\omega}}$	the number of elements in $\boldsymbol{\omega}$
n_i	the number of iterations
n_b	the training batch size
$n_m^{(z)}$	the number of couplings for the z-th attribute
r_{CI}	class imbalance rate
N_o	the set of indices for objects in O
N_a	the set of indices for attributes in A
$N_v^{(j)}$	the set of indices for categorical values in $V^{(j)}$
$\mathscr{M}_{Ia}$	intra-attribute coupling spaces
$\mathscr{M}_{Ie}$	inter-attribute coupling spaces
$\mathscr{K}_p$	the p-th kernel space transformed from a coupling space
$\mathscr{K}_p'$	the heterogeneous kernel space transformed from $\mathscr{K}_p$
$\mathscr{P}$	problem space
$\mathbf{K}_p$	the kernel matrix that spans $\mathscr{K}_p$
$\mathbf{T}_p$	the transformation matrix from $\mathscr{K}_p$ to $\mathscr{K}_p'$
$\mathbf{C}^{(z,k)}$	the k-th coupling matrix of the z-th attribute

α_{pi}	the weight of the i-th value in the p-th kernel space
β_p	the weight of the p-th kernel space
$\mathbf{S}$	similarity representation
$\mathbf{X}$	vector representation
$\mathbf{I}$	The identity matrix

A.2 List of Notations for Dynamic Data

$1,\cdots,t,\cdots,n_t$	The time points from 1 to n_t
E_t	The observed data set at time t
O_t	The set of objects observed at time t
V_t	The set of categorical values observed at time t
$o_{t,i}$	The i-th object in O_t
$V_t^{(j)}$	The set of values of j-th attribute at time t
$v_{t,i}^{(j)}$	The i-th value in the j-th attribute at time t
n_t	The number of time points
n_{o_t}	The number of objects at time t
n_{v_t}	The number of values at time t
$n_{v_t^{(j)}}$	The number of values in the j-th attribute at time t
$n_{v_t^*}$	The number of categorical values represented in the coupling space $\mathcal{M}_{t,j}$ at time t
$\mathbf{x}_{t,i}$	The vector representation of the i-th object at time t
$\mathbf{S}_i^j$	The similarity representation of E_i generated by the models learned at time j
$\mathcal{M}_i^j$	The coupling spaces of observations at time i generated by the model learned at time j
$\mathcal{M}_{Ia,t}^{(j)}$	The intra-attribute coupling space of observations at time t
$\mathcal{M}_{Ie,t}^{(j)}$	The inter-attribute coupling space of observations at time t

$m^{(j)}_{Ia,t}(\cdot)$	The intra-attribute coupling function for the j-th attribute at time t
$m^{(j)}_{Ie,t}(\cdot)$	The inter-attribute coupling function for the j-th attribute at time t
$\mathsf{m}_{t,i}$	A vector of i-th categorical value in a coupling space at time t
$p_t(v^{(j)}\|v^{(k)})$	The information conditional probability function for values $v^{(j)}$ and $v^{(k)}$ at time t
$g^{(j)}_t(v^{(j)})$	A function that returns objects which have value $v^{(j)}$ in j-th attribute at time t
$k_p(\cdot,\cdot)$	The kernel function of the p-th kernel space
$\mathscr{K}_{t,p}$	The p-th kernel space at time t
$\mathscr{K}'_{t,p}$	The p-th heterogeneous kernel space at time t
$\mathbf{K}_{t,p}$	The kernel matrix of the p-th kernel space at time t
$\mathbf{K}'_{t,p}$	The kernel matrix of the p-th heterogeneous kernel space at time t
$\mathbf{T}_{t,p}$	The transform matrix for $\mathbf{K}_{t,p}$ at time t
$\mathbf{S}_t$	The kernel matrix of observed categorical data at time t
$S_{t,p,ij}$	The similarity of the i-th and j-th objects in the p-th heterogeneous coupling space at time t
$S_{t,ij}$	The similarity of the i-th and j-th objects in the heterogeneous coupling space at time t
ξ_t	The coupling difference degree between t and $t-1$
$\alpha_{t,pi}$	The weight of the i-th value in the p-th kernel space at time t
$\beta_{t,p}$	The weight of the p-th base similarity at time t
$\boldsymbol{\omega}_t$	The heterogeneity parameter at time t
$\boldsymbol{\omega}_{t,p}$	The diagonal heterogeneity parameter matrix for the p-th base similarity
$\boldsymbol{\omega}_{t,p,ij}$	The (i,j)-th entry of $\boldsymbol{\omega}_{t,p}$

A.3 List of Abbreviations

AE	Auto-encoder

AOF	Attribute-to-object mapping function
AR	Averaged rank
BMF	Bayesiam matrix factorization
CCDS-DC	Clustering categorical data streams with drifting concepts
CCTED	Clustering categorical time-evolving data
CD	Critical difference
CDE	Coupled data embedding
CI	Class imbalance
cKML	Coupled kernel metric learning
COS	Coupled object similarity
DCD	Clustering concept-drifting categorical data
DILCA	Distance learning of categorical attribute
DNL	Decoupled non-IID learning
ELBO	Evidence lower bound
HC	Hierarchical coupling
HCL	Hierarchical coupling learning
HELIC	Heterogeneous metric learning with hierachical couplings
HL	Heterogeneity learning
HML	Heterogeneity and metric learning
ICPF	Information conditional probability function
IID	Independent and identically distributed
KDE	Kernel density estimation
KDML	Kernel density metric learning
KNN	K-nearest neighbors
LDA	Latent Dirichlet analysis
LR	Logistic regression

MAE	Mean absolute error
MDS	Multidimensional scaling
MTDLE	Multiple transitive distance learning
NB-G	Non-BEND with Gibbs sampling
NB-GU	Non-G without the attribute weighting strategy
NB-V	Non-BEND with VI
NB-VU	NB-V without the attribute weighting strategy
NLP	Natural language processing
NMI	Normalized mutual information
non-BEND	Nonparametric Bayesian embedding
nSML	Non-IID similarity metrics learning
OF	Occurrence frequency
PCA	Principal component analysis
RF	Random forest
RKHS	Reduced kernel Hilbert space
RMSE	Root-mean-square error
SGD	Stochastic gradient descent
SU	Symmetric uncertainty
SVM	Support vector machine
t-SNE	T-distributed stochastic neighbor embedding
UNICORN	Unsupervised hierarchical and heterogeneous coupling learning
UNTIE	Unsupervised heterogeneous coupling learning
VDM	Value difference metric
VFF	Value-frequency-percentage function
VI	Variational inference

BIBLIOGRAPHY

Ahmad, A. & Dey, L. (2007), 'A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set', *Pattern Recognition Letters* **28**(1), 110–118.

Akusok, A., Miche, Y., Hegedus, J., Nian, R. & Lendasse, A. (2014), 'A two-stage methodology using k-nn and false-positive minimizing elm for nominal data classification', *Cognitive Computation* **6**(3), 432–445.

Bai, L., Cheng, X., Liang, J. & Shen, H. (2016), 'An optimization model for clustering categorical data streams with drifting concepts', *IEEE Transactions on Knowledge and Data Engineering* **28**(11), 2871–2883.

Balcan, M.-F., Blum, A. & Srebro, N. (2008), 'A theory of learning with similarity functions', *Machine Learning* **72**(1-2), 89–112.

Barbará, D., Li, Y. & Couto, J. (2002), Coolcat: an entropy-based algorithm for categorical clustering, *in* 'Proceedings of the eleventh international conference on Information and knowledge management', ACM, pp. 582–589.

Battaglia, P. W., Pascanu, R., Lai, M., Rezende, D. J. & Kavukcuoglu, K. (2016), Interaction networks for learning about objects, relations and physics, *in* 'Advances in Neural Information Processing Systems', pp. 4502–4510.

Belkin, M., Niyogi, P. & Sindhwani, V. (2006), 'Manifold regularization: A geometric framework for learning from labeled and unlabeled examples', *The Journal of Machine Learning Research* **7**, 2399–2434.

Bengio, Y., Courville, A. & Vincent, P. (2013), 'Representation learning: A review and new perspectives', *IEEE transactions on pattern analysis and machine intelligence* **35**(8), 1798–1828.

Blei, D. M., Jordan, M. I. et al. (2006), 'Variational inference for dirichlet process mixtures', *Bayesian analysis* **1**(1), 121–143.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003), 'Latent dirichlet allocation', *Journal of machine Learning research* **3**(Jan), 993–1022.

Borg, I. & Groenen, P. J. (2005), *Modern multidimensional scaling: Theory and applications*, Springer Science & Business Media.

Boriah, S., Chandola, V. & Kumar, V. (2008), Similarity measures for categorical data: A comparative evaluation, *in* 'SIAM Conference on Data Mining', SIAM, pp. 243–254.

Brockmeier, A. J., Mu, T., Ananiadou, S. & Goulermas, J. Y. (2017), 'Quantifying the informativeness of similarity measurements', *The Journal of Machine Learning Research* **18**(1), 2592–2652.

Bucak, S. S., Jin, R. & Jain, A. K. (2014), 'Multiple kernel learning for visual object recognition: A review', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(7), 1354–1369.

Cao, F., Liang, J., Bai, L., Zhao, X. & Dang, C. (2010), 'A framework for clustering categorical time-evolving data', *IEEE Transactions on Fuzzy Systems* **18**(5), 872–882.

Cao, F., Liang, J., Li, D., Bai, L. & Dang, C. (2012), 'A dissimilarity measure for the k-modes clustering algorithm', *Knowledge-Based Systems* **26**, 120–127.

Cao, L. (2014), 'Non-iidness learning in behavioral and social data', *The Computer Journal* **57**(9), 1358–1370.

Cao, L. (2015), 'Coupling learning of complex interactions', *Information Processing & Management* **51**(2), 167–186.

Cao, L., Ou, Y. & Yu, P. S. (2012), 'Coupled behavior analysis with applications', *IEEE Transactions on Knowledge and Data Engineering* **24**(8), 1378–1392.

Cao, L., Zhang, H., Zhao, Y., Luo, D. & Zhang, C. (2011), 'Combined mining: discovering informative knowledge in complex data', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **41**(3), 699–712.

Cao, Q., Guo, Z.-C. & Ying, Y. (2016), 'Generalization bounds for metric and similarity learning', *Machine Learning* **102**(1), 115–132.

Chen, H.-L., Chen, M.-S. & Lin, S.-C. (2009), 'Catching the trend: A framework for clustering concept-drifting categorical data', *IEEE transactions on knowledge and data engineering* **21**(5), 652–665.

Cheng, V., Li, C.-H., Kwok, J. T. & Li, C.-K. (2004), 'Dissimilarity learning for nominal data', *Pattern Recognition* **37**(7), 1471–1477.

Cheung, Y.-M. & Jia, H. (2013), 'Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number', *Pattern Recognition* **46**(8), 2228–2238.

Cortes, C., Mohri, M. & Rostamizadeh, A. (2012), 'Algorithms for learning kernels based on centered alignment', *Journal of Machine Learning Research* **13**(Mar), 795–828.

Cuturi, M. & Avis, D. (2014), 'Ground metric learning', *Journal of Machine Learning Research* **15**(1), 533–564.

Demšar, J. (2006), 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine learning research* **7**(Jan), 1–30.

Dhillon, I. S., Guan, Y. & Kulis, B. (2004), Kernel k-means: spectral clustering and normalized cuts, *in* 'Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 551–556.

Duchi, J., Hazan, E. & Singer, Y. (2011), 'Adaptive subgradient methods for online learning and stochastic optimization', *Journal of Machine Learning Research* **12**(Jul), 2121–2159.

Dunson, D. B. & Xing, C. (2009), 'Nonparametric bayes modeling of multivariate categorical data', *Journal of the American Statistical Association* **104**(487), 1042–1051.

Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J. & Zoubin, G. (2013), Structure discovery in nonparametric regression through compositional kernel search, *in* 'International Conference on Machine Learning', pp. 1166–1174.

Frasconi, P., Costa, F., De Raedt, L. & De Grave, K. (2014), 'klog: A language for logical and relational learning with kernels', *Artificial Intelligence* **217**, 117–143.

Gärtner, T., Lloyd, J. W. & Flach, P. A. (2004), 'Kernels and distances for structured data', *Machine Learning* **57**(3), 205–232.

Gönen, M. & Alpaydın, E. (2011), 'Multiple kernel learning algorithms', *Journal of machine learning research* **12**(Jul), 2211–2268.

He, Y., Chen, W., Chen, Y. & Mao, Y. (2013), Kernel density metric learning, *in* 'Data Mining (ICDM), 2013 IEEE 13th International Conference on', IEEE, pp. 271–280.

Huang, G.-B., Zhou, H., Ding, X. & Zhang, R. (2012), 'Extreme learning machine for regression and multiclass classification', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **42**(2), 513–529.

Ienco, D. & Pensa, R. G. (2016), 'Positive and unlabeled learning in categorical data', *Neurocomputing* **196**, 113–124.

Ienco, D., Pensa, R. G. & Meo, R. (2012), 'From context to distance: Learning dissimilarity for categorical data clustering', *ACM Transactions on Knowledge Discovery from Data* **6**(1), 1–25.

Jain, P., Kulis, B., Davis, J. V. & Dhillon, I. S. (2012), 'Metric and kernel learning using a linear transformation', *The Journal of Machine Learning Research* **13**(1), 519–547.

Jegelka, S., Gretton, A., Schölkopf, B., Sriperumbudur, B. K. & Von Luxburg, U. (2009), Generalized clustering via kernel embeddings, *in* 'Annual Conference on Artificial Intelligence', Springer, pp. 144–152.

Jia, H., Cheung, Y.-m. & Liu, J. (2016), 'A new distance metric for unsupervised learning of categorical data', *IEEE transactions on neural networks and learning systems* **27**(5), 1065–1079.

Jian, S., Cao, L., Pang, G., Lu, K. & Gao, H. (2017), Embedding-based representation of categorical data by hierarchical value coupling learning, *in* 'Twenty-Sixth International Joint Conference on Artificial Intelligence', pp. 1937–1943.

Kahane, J.-P. (1993), *Some random series of functions*, Vol. 5, Cambridge University Press.

Kan, M., Shan, S., Zhang, H., Lao, S. & Chen, X. (2015), 'Multi-view discriminant analysis', *IEEE transactions on pattern analysis and machine intelligence* **38**(1), 188–194.

Khorshidpour, Z., Hashemi, S. & Hamzeh, A. (2011), 'Cbdl: Context-based distance learning for categorical attributes', *International Journal of Intelligent Systems* **26**(11), 1076–1100.

Kingma, D. & Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980* .

Kolda, T. G. (2001), 'Orthogonal tensor decompositions', *SIAM Journal on Matrix Analysis and Applications* **23**(1), 243–255.

Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J. & Woźniak, M. (2017), 'Ensemble learning for data stream analysis: A survey', *Information Fusion* **37**, 132–156.

Kullback, S. & Leibler, R. A. (1951), 'On information and sufficiency', *The Annals of Mathematical Statistics* **22**(1), 79–86.

Lazarsfeld, P. F., Henry, N. W. & Anderson, T. W. (1968), *Latent structure analysis*, Vol. 109, Houghton Mifflin Boston.

Le, S. Q. & Ho, T. B. (2005), 'An association-based dissimilarity measure for categorical data', *Pattern Recognition Letters* **26**(16), 2549–2557.

Levy, O. & Goldberg, Y. (2014), Neural word embedding as implicit matrix factorization, *in* 'Advances in neural information processing systems', pp. 2177–2185.

Li, Y., Li, D., Wang, S. & Zhai, Y. (2014), 'Incremental entropy-based clustering on categorical data streams with concept drift', *Knowledge-Based Systems* **59**, 33–47.

Lim, D. & Lanckriet, G. (2014), Efficient learning of mahalanobis metrics for ranking, *in* 'International Conference on Machine Learning', pp. 1980–1988.

Lim, D., Lanckriet, G. R. & McFee, B. (2013), Robust structural metric learning., *in* 'International Conference on Machine Learning', pp. 615–623.

Liu, J. S. (1994), 'The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem', *Journal of the American Statistical Association* **89**(427), 958–966.

Liu, W., Mu, C., Ji, R., Ma, S., Smith, J. R. & Chang, S.-F. (2015), Low-rank similarity metric learning in high dimensions, *in* 'AAAI Conference on Artificial Intelligence', pp. 2792–2799.

Liu, X., Wang, L., Zhu, X., Li, M., Zhu, E., Liu, T., Liu, L., Dou, Y. & Yin, J. (2019), 'Absent multiple kernel learning algorithms', *IEEE transactions on pattern analysis and machine intelligence* .

Maaten, L. v. d. & Hinton, G. (2008), 'Visualizing data using t-sne', *Journal of Machine Learning Research* **9**(Nov), 2579–2605.

Mika, S., Schölkopf, B., Smola, A. J., Müller, K.-R., Scholz, M. & Rätsch, G. (1999), Kernel pca and de-noising in feature spaces, *in* 'Advances in Neural Information Processing Systems', pp. 536–542.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013), Distributed representations of words and phrases and their compositionality, *in* 'Advances in neural information processing systems', pp. 3111–3119.

Narasimhan, H., Pan, W., Kar, P., Protopapas, P. & Ramaswamy, H. G. (2016), Optimizing the multiclass f-measure via biconcave programming, *in* 'IEEE International Conference on Data Mining', IEEE, pp. 1101–1106.

Ng, M. K., Li, M. J., Huang, J. Z. & He, Z. (2007), 'On the impact of dissimilarity measure in k-modes clustering algorithm', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(3), 503–507.

Pang, G., Cao, L. & Chen, L. (2016), Outlier detection in complex categorical data by modelling the feature value couplings, *in* 'Proceedings of the 25th International Joint Conference on Artificial Intelligence', Vol. 2016, pp. 9–15.

Peng, S., Hu, Q., Chen, Y. & Dang, J. (2015), 'Improved support vector machine algorithm for heterogeneous data', *Pattern Recognition* **48**(6), 2072–2083.

Powers, D. & Xie, Y. (2008), *Statistical methods for categorical data analysis*, Emerald Group Publishing.

Qian, Y., Li, F., Liang, J., Liu, B. & Dang, C. (2016), 'Space structure and clustering of categorical data', *IEEE transactions on neural networks and learning systems* **27**(10), 2047–2059.

Ralaivola, L., Szafranski, M. & Stempfel, G. (2010), 'Chromatic pac-bayes bounds for non-iid data: Applications to ranking and stationary β-mixing processes', *The Journal of Machine Learning Research* **11**, 1927–1956.

Rasmussen, C. E. & Williams, C. K. I. (2006), *Gaussian Processes for Machine Learning*, MIT Press.

Rodriguez, A., Dunson, D. B. & Gelfand, A. E. (2008), 'The nested dirichlet process', *Journal of the American Statistical Association* **103**(483), 1131–1154.

Saha, I. & Maulik, U. (2014), 'Incremental learning based multiobjective fuzzy clustering for categorical data', *Information Sciences* **267**, 35–57.

Salakhutdinov, R. & Mnih, A. (2008), Bayesian probabilistic matrix factorization using markov chain monte carlo, *in* 'Proceedings of the 25th international conference on Machine learning', ACM, pp. 880–887.

Shi, Y., Li, W. & Sha, F. (n.d.), Metric learning for ordinal data, *in* 'Thirtieth AAAI Conference on Artificial Intelligence', pp. 2030–2036.

Stanfill, C. & Waltz, D. (1986), 'Toward memory-based reasoning', *Communications of the ACM* **29**(12), 1213–1228.

Steinwart, I. & Christmann, A. (2008), *Support Vector Machines*, Springer Science & Business Media.

Stella, X. Y. & Shi, J. (2003), Multiclass spectral clustering, *in* 'International Conference on Computer Vision', pp. 313–319.

Tutz, G. & Gertheiss, J. (2016), 'Regularized regression for categorical data', *Statistical Modelling* **16**(3), 161–200.

Vapnik, V. N. (1998), *Statistical Learning Theory*, Wiley-Interscience.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. & Manzagol, P.-A. (2010), 'Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion', *Journal of machine learning research* **11**(Dec), 3371–3408.

Wang, C., Cao, L., Wang, M., Li, J., Wei, W. & Ou, Y. (2011), Coupled nominal similarity in unsupervised learning, *in* 'ACM International Conference on Information and Knowledge Management', ACM, pp. 973–978.

Wang, C., Chi, C.-H., Zhou, W. & Wong, R. (2015), Coupled interdependent attribute analysis on mixed data, *in* 'AAAI Conference on Artificial Intelligence', pp. 1861–1867.

Wang, C., Chi, C., She, Z., Cao, L. & Stantic, B. (2018), 'Coupled clustering ensemble by exploring data interdependence', *ACM Transactions on Knowledge Discovery from Data* **12**(6), 63:1–63:38.

Wang, C., Dong, X., Zhou, F., Cao, L. & Chi, C.-H. (2015), 'Coupled attribute similarity learning on categorical data', *IEEE Transactions on Neural Networks and Learning Systems* **26**(4), 781–797.

Wang, C., She, Z. & Cao, L. (2013), Coupled attribute analysis on numerical data, *in* 'International Joint Conference on Artificial Intelligence', AAAI Press, pp. 1736–1742.

Wang, W., Arora, R., Livescu, K. & Bilmes, J. (2015), On deep multi-view representation learning, *in* 'International Conference on Machine Learning', pp. 1083–1092.

Wang, Y., Liu, X., Dou, Y., Lv, Q. & Lu, Y. (2017), 'Multiple kernel learning with hybrid kernel alignment maximization', *Pattern Recognition* **70**, 104–111.

Weinberger, K. Q. & Saul, L. K. (2009), 'Distance metric learning for large margin nearest neighbor classification', *Journal of Machine Learning Research* **10**(Feb), 207–244.

Wu, T.-F., Lin, C.-J. & Weng, R. C. (2004), 'Probability estimates for multi-class classification by pairwise coupling', *J. Mach. Learn. Res.* **5**, 975–1005.

Xie, J., Szymanski, B. K. & Zaki, M. J. (2013), Learning dissimilarities for categorical symbols, *in* 'JMLR: Workshop on Feature Selection in Data Mining', JMLR. org, pp. 2228–2238.

Xie, S., Wang, G., Lin, S. & Yu, P. S. (2012), Review spam detection via temporal pattern discovery, *in* 'Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, pp. 823–31.

Xu, C., Tao, D. & Xu, C. (2015), 'Multi-view intact space learning', *IEEE transactions on pattern analysis and machine intelligence* **37**(12), 2531–2544.

Xue, Y., Liao, X., Carin, L. & Krishnapuram, B. (2007), 'Multi-task learning for classification with dirichlet process priors', *Journal of Machine Learning Research* **8**(Jan), 35–63.

Ye, H.-J., Zhan, D.-C. & Jiang, Y. (n.d.), Instance specific metric subspace learning: A bayesian approach, *in* 'Thirtieth AAAI Conference on Artificial Intelligence', pp. 2272–2278.

Ying, Y. & Li, P. (2012), 'Distance metric learning with eigenvalue optimization', *Journal of Machine Learning Research* **13**(Jan), 1–26.

Yu, S., Tranchevent, L., Liu, X., Glanzel, W., Suykens, J. A., De Moor, B. & Moreau, Y. (2012), 'Optimized data fusion for kernel k-means clustering', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **34**(5), 1031–1039.

Zhang, K., Wang, Q., Chen, Z., Marsic, I., Kumar, V., Jiang, G. & Zhang, J. (2015), From categorical to numerical: Multiple transitive distance learning and embedding, *in* 'SIAM International Conference on Data Mining', SIAM, pp. 46–54.

Zhang, Q., Cao, L., Zhu, C., Li, Z. & Sun, J. (2018), Coupledcf: Learning explicit and implicit user-item couplings in recommendation for deep collaborative filtering., *in* 'IJCAI', pp. 3662–3668.

Zhao, B., Kwok, J. T. & Zhang, C. (2009), Multiple kernel clustering, *in* 'Proc. of the SDM'09', SIAM, pp. 638–649.

Zhao, X., Zhu, C. & Cheng, L. (2017), Coupled bayesian matrix factorization in recommender systems, *in* 'IEEE International Conference on Data Science and Advanced Analytics', IEEE, pp. 522–528.

Zhu, C., Cao, L., Liu, Q., Yin, J. & Kumar, V. (2018), 'Heterogeneous metric learning of categorical data with hierarchical couplings', *IEEE Transactions on Knowledge and Data Engineering* **30**(7), 1254–1267.