

“© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

CURE: Flexible Categorical Data Representation by Hierarchical Coupling Learning

Songlei Jian, Guansong Pang, Longbing Cao, *Senior Member, IEEE*, Kai Lu, Hang Gao

Abstract—The representation of categorical data with hierarchical coupling relationships (i.e., value to value cluster interactions) is very critical yet challenging for capturing data characteristics in learning tasks. This paper proposes a novel and flexible coupled unsupervised categorical data representation (CURE) framework which not only captures the hierarchical couplings but also is flexible to be instantiated for contrastive learning tasks. CURE first learns value clusters with different granularities based on multiple value coupling functions and then learns value representation from the couplings between the obtained value clusters. Based on two complementary value coupling functions, CURE is instantiated into two instances: the coupled data embedding (CDE) for clustering and the coupled outlier scoring of high-dimensional data (COSH) for outlier detection, by customizing the ways of value clustering and coupling learning between value clusters. CDE embeds categorical data into a new space in which features are independent and semantics are rich. COSH represents data with an outlying vector to capture complex outlying behaviors of objects in high-dimensional data. Substantial experiments show that CDE significantly outperforms three popular unsupervised embedding methods (with maximum 19% improvement) and three state-of-the-art similarity-based representation methods (maximum improvement of 8%), and COSH performs significantly better than five state-of-the-art outlier detection methods (with 67% maximum improvement) on high-dimensional data sets. CDE and COSH are scalable and stable, linear to data size and quadratic to the number of features, and are insensitive to their parameters.

Index Terms—Categorical Data Representation, Unsupervised Learning, Coupling Learning, Clustering, Outlier Detection.

1 INTRODUCTION

CATEGORICAL data with finite unordered feature values is ubiquitous in real-world applications and has received increasing attention for representation and learning [1], [2]. Unlike numerical data, categorical data cannot be directly manipulated per algebraic operations; hence many popular numerical learning algorithms are not directly applicable. Accordingly, it is important to learn an expressive numerical representation of categorical data.

1.1 Motivations

In general, a good representation should effectively capture the intrinsic data characteristics [3]. One key characteristic in complex categorical data is the following hierarchical couplings (i.e., different types of interactions) embedded in feature values. (1) On the low level, there exist strong couplings [4] between feature values, demonstrating the natural clustering of values. Taking the census data as an example, it may be visible that the value *PhD* of feature *Education* is highly coupled with the values *Scientist* and *Professor* of feature *Occupation*; and these values form a semantic value cluster that characterizes one type of strong

relation between education and occupation. In addition, different value clusters exist on different granularities and with different semantics [5]; e.g., all values belong to one super cluster at the coarsest granularity while each value is a cluster at the finest granularity. (2) On the high level, the clusters of feature values are further coupled with each other. Couplings exist between clusters of the same granularity and between clusters of different granularities.

A good representation should enable learning tasks to obtain better learning performance. For unsupervised learning, there are two main tasks: clustering and outlier detection (or anomaly detection). Clustering aims to assign all objects to different clusters, so the clustering performance is decided by the majority of data objects; while outlier detector aims to find abnormal objects (i.e., rare or inconsistent objects compared to the majority of objects), so the performance is decided by the minority of data objects.

For clustering, the more relevant information the representation captures, the more reliable the clustering is, especially for complex data where there are hierarchical couplings. However, existing embedding and similarity-based representation methods for clustering can capture only a part or none of these feature value couplings. Typical embedding-based representation methods transform categorical data to numerical data by encoding schemes, e.g., one-hot encoding and Inverse Document Frequency (IDF) encoding [6]. These methods are easy to implement, but do not consider the couplings between feature values since they usually treat features independently. Some recent similarity-based representation methods, e.g., in [1], [7], [8], [9] incorporate feature relations into similarity or kernel matrices. However, they do not capture the couplings from values to value clusters or couplings between value clusters, leading

- Songlei Jian is with the Laboratory of Science and Technology on Parallel and Distributed Processing and the College of Computer, National University of Defense Technology, China. Songlei Jian is also visiting the Advanced Analytics Institute, University of Technology Sydney, Australia.
- Guansong Pang and Longbing Cao are with the Advanced Analytics Institute, University of Technology Sydney, Australia.
- Hang Gao and Kai Lu is with the Laboratory of Science and Technology on Parallel and Distributed Processing and the College of Computer, National University of Defense Technology, China.

Manuscript received April 19, 2005; revised August 26, 2015. (Corresponding authors: Longbing Cao.)

to insufficient representation power in handling data with such hierarchical value couplings.

For outlier detection, the representation capturing more relevant information, however, does not guarantee better performance. The captured information also needs to be outlier-discriminative. Most encoding or similarity-based methods [1], [7], [8] are majority-based representation approaches, which do not capture the abnormal aspects of the data. Different from these methods, most existing outlier detection methods for categorical data [10], [11], [12], [13] use pattern-based representation (i.e., the data is represented by a set of outlying/normal patterns) to disclose the characteristics of outliers. However, the patterns are normally a subset of compact predefined value combinations and can only capture partial couplings between values. This may result in less expressive representation power in data with sophisticated value couplings, in particular high-dimensional data, in which there exists a complex mixture of relevant features and irrelevant features. A very recent method called CBRW [14] models the full value couplings to generate value-based representation for categorical outlier detection. The value-based representation is more fine-grained and flexible than the pattern-based methods. However, CBRW captures only pair-wise value interactions and thus fails to work in data sets with high-order interactions between values.

1.2 Contributions

Therefore, the hierarchical value to value cluster couplings, which reflect some key intrinsic data characteristics and complexities, need to be captured and properly customized in data representations for different learning tasks and application scenarios. However, this is not a trivial task and, to our best knowledge, no work reported properly handles them. Accordingly, this paper aims to propose a flexible framework which captures the hierarchical value couplings and could be instantiated to solve different problems. The main idea and contributions are as follows.

- A Coupled Unsupervised categorical data REpresentation (CURE for short) framework is proposed, which has a hierarchical learning structure and is flexible to be instantiated. CURE defines multiple value coupling functions for clustering values with different granularities to capture the low-level complex couplings between values. CURE further learns couplings between the multi-granularity value clusters to incorporate high-order interactions between values into our value-based data representation. This enables CURE to capture the intrinsic data characteristics and produce an effective numerical representation for categorical data with sophisticated couplings.
- For clustering, CURE is instantiated into a Coupled Data Embedding (CDE for short) method to capture majority-based hierarchical value couplings. CDE utilizes the couplings to embed categorical data into a new space with independent dimensions and rich semantics. This creates a meaningful Euclidean space for the subsequent object clustering.
- For outlier detection, CURE is instantiated into a Coupled Outlier Scoring of High-dimensional data (COSH for short) method to capture minority-based hierarchical value couplings. COSH uses the multi-granularity

value clusters to compute the most outlying aspect of values, which enables it to obtain reliable outlier scores in data sets with many irrelevant/noisy features.

Substantial experiments show that (1) CDE significantly outperforms three popular embedding methods: one-hot encoding (noted as 0-1), one-hot encoding with PCA (0-1P), and inverse document frequency embedding (IDF), with maximum F-score improvement of 19%, and gains maximally 8% F-score improvement over three state-of-the-art similarity measures for clustering: COS [1], DILCA [8] and ALGO [7] on 10 real-world data sets with different value coupling complexities; (2) COSH significantly outperforms (maximally 67% AUC improvement) five state-of-the-art outlier detection methods: CBRW [14], and ZERO [15], iForest [16], ABOD [17] and LOF [18] on 10 high-dimensional data sets; (3) CDE and COSH obtain good scalability: it is linear to data size and quadratic to the number of features; and (4) CDE and COSH perform stably and are insensitive to its parameters.

The rest of this paper is organized as follows. We discuss the related work in Section 2. The CURE framework are detailed in Section 3. Two complementary value coupling functions are presented in Section 4. The two instances of CDE and COSH are introduced in Section 5. Experimental results for clustering and outlier detection are provided in Section 6 and Section 7 respectively. Discussion of instantiating CURE is given in Section 8. Conclusion are shown in Section 9.

2 RELATED WORK

This section discusses representation methods for the clustering and outlier detection of categorical data. The representation methods for clustering consists of two main parts: embedding-based and similarity-based methods, while representation for outlier detection is scoring-based methods.

2.1 Representation for Clustering

Embedding-based Representation. Embedding-based representation constructs a numerical vector for each categorical object. Encoding methods are the most widely used for categorical data representation [19]. One popular method is the one-hot encoding which encodes each feature with a zero-one matrix. Feature f_i is encoded with $|\mathcal{V}_i|$ -dimensional vectors, where each vector has a single 1 corresponding to one value, and all rest entries are 0s. Although one-hot coding is reversible with the original data, it assumes that all values are independent and equal which is often violated in real-world data. Also, one-hot encoding results in very high dimensions if original data has a large number of values, and consequently it may lead to the curse of dimensionality issue [20]. Dimension reduction methods, like principal component analysis (PCA) [21], are often conducted on a one-hot encoding matrix to alleviate the issue. Another well-known method is IDF encoding [6] which represents each value with the logarithm of its inverse frequency. IDF captures the value couplings from occurrence perspective. Although these simple embedding methods are easy to implement and have a good efficiency, they cannot capture the complex value couplings in data. CDE captures the

hierarchical couplings between values and value clusters and keeps a good efficiency at the same time.

Several effective embedding methods are available for textual data, such as latent semantic indexing (LSI) [22], latent Dirichlet allocation (LDA) [23], skip-gram [24] and their variants [25], [26], [27]. However, categorical data has an explicit feature structure, which is very different from unstructured textual data. Hence, these methods do not fit our target problem.

Similarity-based Representation. Similarity-based representation approaches (including some kernel methods) represent categorical data with an object similarity matrix. Various similarity measures have been designed to capture value couplings in data: ALGO [7] first use conditional probability of two feature values to describe the value couplings; DILCA [8] and DM [9] incorporate feature selection and feature weighting into capturing feature couplings respectively; COS [1] takes inter- and intra-feature couplings into object similarity. Similar to COS, coupledMC [28] transforms categorical feature value into real value which is calculated according to the similarity between values. These similarity measures focus on capturing the pairwise value couplings. They therefore fail to capture the couplings among multiple values and higher order interactions, which are captured by CDE in the couplings between value clusters. Meanwhile, similarity measurement is not an efficient method of representation since it needs to calculate and store the pairwise similarity. The storage complexity is at least quadratic w.r.t. the number of objects which may limit its applications on large data. In contrast, CDE has linear time complexity w.r.t. data size and well scales up to large-scale data.

In addition, there are some embedding methods, e.g., the one in [29], [30] which optimizes the embedding representation on the similarity matrix, but their results heavily rely on the underlying similarity measures. Some other embedding methods (e.g., [2], [31]) require class labels to learn distance, and thus they are inapplicable for unsupervised tasks.

2.2 Representation for Outlier Detection

Most existing outlier detection methods [10], [11], [12], [13] for categorical data unify the two successive tasks - data representation and outlier identification. These methods often aim to identify a set of outlying/normal patterns to represent the data objects. Since the identified patterns carry the outlieriness of objects, they can be directly used to identify outliers. Such outlier detection-oriented methods are referred to as scoring-based representation approaches, since they are very different from embedding- or similarity-based representation which separates model learning from data representation learning and focuses on how to effectively transform the original data into a meaningful space for well enabling subsequent model learning. However, these methods involve costly pattern discovery. As a result, their computational time is prohibitive in high-dimensional data. Also, these methods become ineffective in handling data sets with many irrelevant/noisy features [14].

There have been some methods (e.g., in [14], [15], [32]) scalable to high-dimensional data. The method CBRW [14] models the intra- and inter-feature value couplings to estimate the outlieriness of values and uses the value outlieriness

to represent the objects. CBRW is a closely related work to COSH as it also attempts to use value outlieriness to represent the data. CBRW avoids the costly pattern search and has good scalability w.r.t. the data dimensionality. However, CBRW only captures pair-wise value interactions and may fail to work in data with higher order value interactions, e.g., high-dimensional data, while COSH captures such higher order interactions in the couplings between value clusters. The method ZERO++ [15] can efficiently handle high-dimensional data by working on a random set of feature subspaces, but the random subspace generation may include many irrelevant features and downgrade its performance in those data. The method ITB [32] aims to identify a set of outliers so that the removal of these outliers from the data set mostly reduce the entropy-based data uncertainty. It has linear time complexity w.r.t. the number of features. However, it uses the full feature sets to compute the uncertainty and is largely affected by the irrelevant features, and thus it becomes less effective in high-dimensional data where outliers are manifested in a small subset of features.

Some methods like ABOD [17], [33] and iForest [16] for high-dimensional numeric data may also be extended to handle categorical data by working on its embedding- or similarity-based numeric representation, but their performance is heavily dependent on the effectiveness of the data representation methods.

More importantly, all the above methods estimate the outlier scores based on single-granularity outlieriness representation, i.e., the outlieriness estimation operates with the same granularity; whereas our method COSH captures the outlieriness with a wide range of granularity. Our outlieriness estimation is therefore less likely to be biased by the overwhelming irrelevant features in high-dimensional data.

3 THE CURE FRAMEWORK FOR CATEGORICAL DATA REPRESENTATION

In this paper, we introduce the CURE framework which models hierarchical couplings between values and value clusters to learn a numerical representation of categorical data. As shown in Fig. 1, CURE first learns the low-level couplings between values by several coupling functions. Then it learns value clusters with different granularities by clustering on multiple value coupling matrices with different granularity settings. CURE further learns the couplings between value clusters to obtain the value representation and the object representation.

Let $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$ be a set of data objects with size N , described by a set of D categorical features $\mathcal{F} = \{f_1, \dots, f_D\}$. Each feature f ($f \in \mathcal{F}$) has a value domain $\mathcal{V}_f = \{v_1, v_2, \dots\}$ which consists of a finite set of possible feature values (at least two values). The value domains of different features are distinct, i.e., $\mathcal{V}_{f_i} \cap \mathcal{V}_{f_j} = \emptyset, \forall i \neq j$. The whole value set of features is the union of all the value domains: $\mathcal{V} = \cup_{f \in \mathcal{F}} \mathcal{V}_f$, and the size of \mathcal{V} is denoted as L .

Problem Statement Our problem can then be stated as follows. Given a set of data objects \mathcal{X} , we aim to learn the object numerical representation \mathbf{O} of \mathcal{X} . Following the process of CURE framework, we firstly construct the value coupling set $\Phi(\mathcal{X})$ by learning value couplings. Secondly we learn the value clusters in the value clustering process Ω_η .

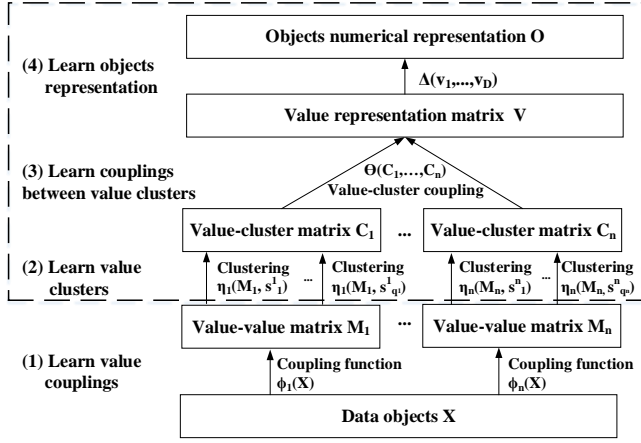


Fig. 1. The CURE framework in which Φ , η , Θ and Δ can be customized according to different tasks. By changing the dashed line boxed part, we instantiate the framework into two instances: CDE and COSH.

Thirdly, the couplings between value clusters are learned in coupling learning process Θ . Finally, the object representation are learned by Δ . The four components of CURE: Φ , Ω_η , Θ and Δ are introduced in details in the next four subsections.

3.1 Learning Value Couplings

Value couplings refer to the interactions of feature values which may include the interactions between values from the same feature and interactions between values from different features. Such value couplings reflect the low-level interactions between values. The more value couplings learned will benefit more to the following value clusters. The definition of the value coupling set is given as follows.

Definition 1 (Value Coupling Set). *The value coupling set $\Phi(\mathcal{X})$ is defined as a set of multiple value coupling functions with size of n to capture the low level pairwise value couplings:*

$$\Phi(\mathcal{X}) = \{\phi_i(\mathcal{X}), i = 1, 2, \dots, n\}, \quad (1)$$

where $\phi_i(\cdot) : \mathcal{X} \mapsto \mathbf{M}_i \in \mathbb{R}^{L \times L}$ is one kind of value coupling functions, to capture the value couplings from one specific perspective. The output of ϕ_i is a value coupling matrix \mathbf{M}_i which consists of couplings between each value pair.

These value coupling matrices are decided by the value coupling functions and reflect the low-level data characteristics. The value coupling functions can be specified from several aspects, e.g., occurrence-based and co-occurrence-based functions, set theory-based functions (such as intersection of value sets), value neighbourhood-based functions and/or non-co-occurrence-based functions. Good value coupling functions should capture different kinds of couplings.

3.2 Learning Value Clusters

A value cluster refers to the value set which consists of multiple similar values. The value clusters reflect the couplings among multiple values instead of pairwise value coupling, e.g., all values belong to one super value cluster at

the coarsest granularity while each value is a cluster at the finest granularity. The definition of value clustering process is given as follows.

Definition 2 (Value Clustering Process). *The value clustering process w.r.t. value coupling matrix \mathbf{M} consists of multiple clustering on value couplings matrices with different granularities which is defined as follows:*

$$\Omega_\eta = \{\eta_i(\mathbf{M}_i, s_j^i), j = 1, 2, \dots, q^i\}, \quad (2)$$

where η_i is one clustering process on value coupling matrix \mathbf{M}_i , and s_j^i is the clustering parameter which decides the granularity of clusters. The output of η_i is a value cluster matrix $\mathbf{C}_i \in \mathbb{R}^{L \times q^i}$.

The value clustering process can be done by various clustering methods, e.g., centroid-based clustering algorithm, hierarchical clustering algorithms, distribution-based clustering, and density-based clustering algorithms. The granularities of value clusters can be decided by the pre-defined algorithm parameters, e.g., the cluster number, and the density range parameter. Different clustering algorithms prefer different kinds of clusters. For example, centroid-based clustering algorithms capture the convex shape of clusters, while density-based clustering algorithms are able to capture the manifold shape of clusters. We can conduct different clustering algorithms on different value coupling matrices or conduct only one clustering algorithm on all coupling matrices with different parameters. The choice of clustering process is decided by the clusters' characteristics captured by the clustering algorithm and its efficiency.

3.3 Learning Couplings between Value Clusters

The value clusters learned by clustering may contain couplings and redundancy in them. Through learning the complex couplings between value clusters, CURE learns the meaningful value representation. The definition of coupling learning between value clusters is defined as follows.

Definition 3 (Coupling Learning Between Value Clusters). *The coupling learning process Θ between value clusters is defined as follows:*

$$\mathbf{V} = \Theta\{\mathbf{C}_1, \dots, \mathbf{C}_n\}, \quad (3)$$

where \mathbf{C}_i is one value cluster matrix and $\mathbf{V} \in \mathbb{R}^{L \times \sum_{i=1}^n q^i}$ is the value representation matrix.

The coupling learning process between value clusters aims to learn the correlation or interaction of different value clusters and try to eliminate the redundancy information among value clusters. So Θ can be implemented by a dimensionality reduction process, relation learning process, or an embedding model, e.g., principal component analysis (PCA), linear discriminant analysis (LDA), matrix factorization and neural network. The choice of Θ depends on the data characteristics and the subsequent learning tasks.

3.4 Learning Object Representation

Based on the value representation, we model the object representation learning process.

Definition 4 (Object Representation Learning Function). *The representation of an object x ($x \in \mathcal{X}$) is modelled by an incorporating function w.r.t. value representation \mathbf{V} :*

$$\mathbf{O}^x = \Delta(\mathbf{V}_1^x, \dots, \mathbf{V}_D^x), \quad (4)$$

where \mathbf{V}_i^x is the value representation of object x from feature f_i .

The object representing function $\Delta(\cdot)$ utilizes value representation to assign each object a numerical vector as representation, which could be specified according to the application of instances, e.g., concatenation, weighted sum, or maximum.

4 COMPLEMENTARY VALUE COUPLINGS

In this paper, we instantiate the CURE framework into two instances: CDE for clustering and COSH for outlier detection according to their different learning goals. Both CDE and COSH are based on the same value coupling functions which is the basis for further value clusters learning. In this section, we introduce the two value coupling function and prove its complementary discriminative ability.

4.1 Definition of Two Value Coupling Functions

To learn value couplings, we construct two value influence matrices to capture the value couplings from two basic perspectives: occurrence and co-occurrence, whose complementary discriminative ability is proved in Section 4.2. Before introducing the value influence matrices, we introduce some preliminaries.

The value from feature f of object x is denoted by v_x^f and the feature which the value v_i belongs to is denoted by f_i . We assume that the probability $p(v)$ of a value can be computed by its frequency. The joint probability of two values v_i and v_j is $p(v_i, v_j) = \frac{|\{v_x^{f_i} = v_i \cap v_x^{f_j} = v_j, x \in \mathcal{X}\}|}{N}$.

We define normalized mutual information [34] ψ to reflect the relation between two features as follows:

$$\psi(f_a, f_b) = \frac{2 \sum_{v_i \in \mathcal{V}_{f_a}} \sum_{v_j \in \mathcal{V}_{f_b}} p(v_i, v_j) \log \frac{p(v_i, v_j)}{p(v_i)p(v_j)}}{h(f_a) + h(f_b)}, \quad (5)$$

where $h(f_a) = -\sum_{v_i \in \mathcal{V}_{f_a}} p(v_i) \log(p(v_i))$.

Definition 5 (Occurrence-based Value Influence Matrix). *The occurrence-based value influence matrix \mathbf{M}_o is defined as follows:*

$$\mathbf{M}_o = \begin{bmatrix} \phi_o(v_1, v_1) & \dots & \phi_o(v_1, v_L) \\ \vdots & \ddots & \vdots \\ \phi_o(v_L, v_1) & \dots & \phi_o(v_L, v_L) \end{bmatrix}, \quad (6)$$

where the coupling function $\phi_o(v_i, v_j) = \psi(f_i, f_j) \times \frac{p(v_j)}{p(v_i)}$ indicates the occurrence influence on value v_i from value v_j .

The occurrence or marginal probability is the basic univariate property of values, which can be used to differentiate values. Instead of symmetric distance measure between the marginal probabilities of two values, we use an asymmetric ratio to quantify the influence on one value from another value so that \mathbf{M}_o captures more information. Besides, we incorporate mutual information ψ as the weight of value

couplings since marginal probabilities cannot differentiate different features.

Definition 6 (Co-occurrence-based Value Influence Matrix). *The co-occurrence-based value influence matrix \mathbf{M}_c is defined as follows:*

$$\mathbf{M}_c = \begin{bmatrix} \phi_c(v_1, v_1) & \dots & \phi_c(v_1, v_L) \\ \vdots & \ddots & \vdots \\ \phi_c(v_L, v_1) & \dots & \phi_c(v_L, v_L) \end{bmatrix}, \quad (7)$$

where the coupling function $\phi_c(v_i, v_j) = \frac{p(v_i, v_j)}{p(v_i)}$ indicates the co-occurrence influence on value v_i from value v_j .

The co-occurrence or joint probability is the basic bivariate relation between two values, which captures the most natural dependency between two variables. We use asymmetric conditional probability to define the influence on one value from another value since the same joint probability may have different influence on values with different marginal probabilities. The ϕ_c value of two values from the same feature always equals 0 since they never co-occur in the same object.

4.2 Complementary Discriminative Ability

The two coupling functions is complementary and discriminative for the values which can be verified by the distance of \mathbf{M}_o and \mathbf{M}_c . Since the following instances CDE and COSH learn value clusters with k -means clustering which is based on the Euclidean distance, we thus take the Euclidean distance as an example to show complementary discriminative ability of the above two value coupling functions.

The distance matrix in k -means clustering determines the quality of value clusters. By proving the complementary discriminative ability of the two distance matrices, we can observe that the two value couplings have complementary discriminative ability.

The occurrence distance between values v_i and v_j is defined as follows:

$$d_o(v_i, v_j) = \sqrt{\sum_{h=1}^L (\phi_o(v_i, v_h) - \phi_o(v_j, v_h))^2}, \quad (8)$$

where $\phi_o(v_i, v_h)$ is the occurrence coupling function defined in Definition 5, and L is the number of values.

The co-occurrence distance between values v_i and v_j is defined below:

$$d_c(v_i, v_j) = \sqrt{\sum_{h=1}^L (\phi_c(v_i, v_h) - \phi_c(v_j, v_h))^2}, \quad (9)$$

where $\phi_c(v_i, v_h)$ is the co-occurrence coupling function defined in Definition 6. If any two distinct values can be distinguished by d_o or d_c , then d_o and d_c are complementary.

Theorem 1 (Distance Complementarity). *For any two values $v_i \neq v_j$, $d_o(v_i, v_j) \neq 0$ or $d_c(v_i, v_j) \neq 0$.*

Proof. To prove the above theorem, we prove that $v_i \neq v_j$ and $d_o(v_i, v_j) = 0$ satisfy $d_c(v_i, v_j) \neq 0$ for all cases and $v_i \neq v_j$ and $d_c(v_i, v_j) = 0$ satisfy $d_o(v_i, v_j) \neq 0$ for all cases.

We first prove that $v_i \neq v_j$ and $d_o(v_i, v_j) = 0$ satisfy $d_c(v_i, v_j) \neq 0$ for all cases. If $d_c(v_i, v_j) = 0$, then $\forall v_h \in$

$\mathcal{V}, \phi_c(v_i, v_h) = \phi_c(v_j, v_h)$. To prove $d_c(v_i, v_j) \neq 0$, we only need to prove $\exists v_h \in \mathcal{V}, \phi_c(v_i, v_h) \neq \phi_c(v_j, v_h)$. We consider the proof for the following cases.

(1) If v_i and v_j belong to the same feature which means $\psi(f_i, f_h) = \psi(f_j, f_h)$, then $d_o(v_i, v_j) = 0$ if and only if $p(v_i) = p(v_j)$. Let $v_h = v_i$, then $\phi_c(v_i, v_h) = 1$ and $\phi_c(v_j, v_h) = 0$ since v_i, v_j belong to the same feature. Hence, $d_c(v_i, v_j) \neq 0$ when v_i and v_j belong to the same feature.

(2) If v_i and v_j belong to different features, and $d_o(v_i, v_j) = 0$ which means $\forall v_h \in \mathcal{V}, \psi(f_i, f_h) \frac{p(v_h)}{p(v_i)} = \psi(f_j, f_h) \frac{p(v_h)}{p(v_j)}$; When $\psi(f_i, f_h) \neq \psi(f_j, f_h)$ and $p(v_i) \neq p(v_j)$ (suppose $p(v_i) < p(v_j)$), then $p(v_i, v_j) < p(v_j)$. Let $v_h = v_i$, then $\phi_c(v_i, v_h) = 1$ and $\phi_c(v_j, v_h) < 0$. Accordingly, $d_c(v_i, v_j) \neq 0$ when $p(v_i) \neq p(v_j)$. When $\psi(f_i, f_h) = \psi(f_j, f_h)$ and $p(v_i) = p(v_j)$, $\exists v_h$ in feature f_i and $p(v_j, v_h) > 0$, but $p(v_i, v_h) = 0$, then $\phi_c(v_j, v_h) \neq \phi_c(v_i, v_h)$. Therefore, $d_c(v_i, v_j) \neq 0$ when v_i and v_j belong to different features.

Further, we prove $v_i \neq v_j$ and $d_c(v_i, v_j) = 0$ satisfy $d_o(v_i, v_j) \neq 0$ for all cases. We consider the proof for the following cases.

(1) If v_i and v_j belong to the same feature, then we could let $v_h = v_i$ so that $\phi_c(v_i, v_h) = 1$ and $\phi_c(v_j, v_h) = 0$. Then we can prove that $d_o(v_i, v_j) \neq 0$.

(2) If v_i and v_j belong to different features, then we could consider $p(v_i) = p(v_j)$ or $p(v_i) \neq p(v_j)$. If $p(v_i) = p(v_j)$ and $d_c(v_i, v_j) = 0$, then $\psi(f_i, f_h) = 1$ which is impossible for different features. Otherwise, we could let $v_h = v_i$ (suppose $p(v_i) < p(v_j)$) then $\phi_c(v_i, v_h) = 1$ and $\phi_c(v_j, v_h) < 0$, and $d_c(v_i, v_j)$ cannot be 0. So if $d_c(v_i, v_j) = 0$, then v_i and v_j must belong to the same feature. \square

The theorem shows that the two value couplings are able to distinguish any two different values. For clustering, the theorem says that at least one clustering process is able to differentiate any two values in extreme case where each value belongs to one cluster. For outlier detection, the theorem states that the outlier detector could differentiate the outlying behavior between any two values. For different applications, we can enhance the discriminative ability from a specific aspect by utilizing different information of value clusters. The following section demonstrates how to utilize the value couplings to learn value clusters and couplings between value clusters to achieve different goals.

5 TWO CONTRASTIVE INSTANCES

In this section, we show two instances of CURE: CDE for clustering and COSH for outlier detection in high-dimensional data. CDE and COSH use the above value couplings, but they use different methods to learn value clusters and couplings between value clusters.

5.1 CDE: The CURE Instance for Clustering

In this subsection, we introduce an instance CDE for clustering which aims to capture the couplings among majority values based on the above value couplings. CDE learns the value clusters with different granularities by multiple k -means clustering with different cluster numbers k . By filtering the value clusters which have less discriminative

information for majority values, CDE differentiates values according to the value to value cluster affiliation. Based on the information in filtered value clusters, CDE learns the couplings between value clusters with PCA. The object embedding is the concatenation of value representation.

5.1.1 Learning Value Clusters for Clustering

Based on the two value influence matrices, we can learn the value clusters with different granularities which represent different semantics and well reflect the data characteristics. To learn the value clusters with different granularities, here we conduct clustering on the value matrices with different cluster numbers.

We conduct k -means clustering on \mathbf{M}_o with different k , i.e., $\{k_1, k_2, \dots, k_{n_o}\}$, and on \mathbf{M}_c with $\{k_1, k_2, \dots, k_{n_c}\}$. The clustering results are represented by a cluster membership indicator matrix \mathbf{C}^I which is defined as follows:

$$\mathbf{C}^I(i, j) = \begin{cases} 1 & \text{if } v_i \text{ is in cluster } j, \\ 0 & \text{if } v_i \text{ is not in cluster } j. \end{cases} \quad (10)$$

For the majority values, the value cluster with a small number of values has less discriminative information since CDE aims to generate the value clusters which can differentiate more values. Accordingly, we remove the small value clusters which only have one value. k is also decided by the removed small clusters which will be discussed in Section 5.1.3. We further concatenate the two indicator matrices derived from two value influence matrices and get a large indicator matrix to represent each value whose dimensionality is no more than $(\sum_{i=1}^{n_o} k_i + \sum_{j=1}^{n_c} k_j)$.

k -means clustering is chosen for two major reasons: (1) The value influence matrices are numerical and the Euclidean distance fed in k -means clustering captures the global relation between values. (2) k -means clustering is linear w.r.t. the size of the input matrix, which enables CDE to efficiently learn value clusters with different sizes.

5.1.2 Learning Linear Couplings between Value Clusters

The indicator matrix \mathbf{C}^I conveys rich couplings between the value clusters with different granularities based on two value influence matrices. For simplicity and the consideration of common scenarios, we assume that couplings between value clusters are linear correlations, and apply PCA on the indicator matrix to eliminate the linear correlations between value clusters to obtain a vector embedding for each value. PCA is chosen because (1) it reduces the data complexity with little loss of information by converting a matrix with linearly correlated variables to a new matrix with linearly uncorrelated components, and (2) it substantially reduces the dimensionality of the value embedding, which enables us to represent an object in a considerably lower-dimensional embedding space.

We first calculate the centralized matrix \mathbf{Z} of the indicator matrix \mathbf{C}^I by subtracting the mean of each column and further derive a covariance matrix \mathbf{S} from \mathbf{Z} . The value embedding \mathbf{V} is obtained by the following matrix decomposition:

$$\mathbf{V} = \mathbf{Z}\mathbf{Y}^T, \quad (11)$$

where \mathbf{Y} is the principal component matrix derived from the singular value decomposition results of \mathbf{S} , i.e., $\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{Y}$.

After the PCA transformation, the dimensions of value embedding \mathbf{V} are independent of each other so that the algebraic operations in the Euclidean space can be used on the embedded matrix.

5.1.3 The CDE Algorithm

Algorithm 1 presents the main procedures of CDE. The first step is to generate the value influence matrices \mathbf{M}_o and \mathbf{M}_c according to Definitions (5) and (6) by scanning the original data matrix. Specifically, we scan the data matrix by rows. For each row, we scan it by columns for two times, and then we can get the co-occurrence of any two values. After scanning all rows we can calculate the frequency of any value. Then we can easily calculate the coupling functions.

k is the clustering parameter which decides the granularity of value clusters. Instead of setting to a fixed value, we use another proportion factor α to decide the maximum cluster number, as shown in Steps (6-10) of Algorithm 1. The clusters that only have one value are meaningless to value cluster. Therefore, we remove those small clusters with only value by controlling the proportion of small clusters through α . With the increasing of k more small clusters are generated. Until the proportion of removed small clusters, i.e., $\frac{length(rm)}{k}$, exceeds the α , we stop increasing k whose initial value is 2. The final \mathbf{C}^I is the concatenation of all clustering results with different k from \mathbf{M}_o and \mathbf{M}_c .

After conducting PCA on the indicator matrix to learn the correlations between value clusters, we treat \mathbf{V} as the original representation of values where each column represents a dimension. Since the distance between two values is the sum of the distance on each dimension, the columns with small range have less contribution to the final distance. We remove those columns whose range (maximum element minus minimum element) is less than β from original representation \mathbf{V} . In this way, we control the dimension of the representation vectors through a flexible data-dependent way. Finally, we calculate the object embedding \mathbf{O} by concatenating embedding vectors of its values from \mathbf{V} .

We generate the \mathbf{M}_o and \mathbf{M}_c through value frequency vector and co-occurrence matrix. Scanning the data set and count the frequency of all values and co-occurrence of all value pairs have the complexity of $O(ND^2)$. Generating \mathbf{M}_o and \mathbf{M}_c based on frequency vector and co-occurrence matrix have the complexity of $O(L^2)$. The total number of clustering times is $(k_{max} - 1)$ due to that k_{max} increases from 2. Then clustering on the value matrix has complexity $O(k_{max}L)$ since k-means clustering has linear complexity w.r.t. the size of the input matrix. The number of value clusters are proportional to k_{max}^2 and then PCA has $O(k_{max}^6)$. With the numerical representation of values, generating the embedding matrix of objects has $O(ND)$. The computational complexity of CDE is $O(ND^2 + L^2 + k_{max}^6)$. Since k_{max} does not increase w.r.t. the D and N and k_{max} is a relatively small constant, k_{max}^6 is much smaller than ND^2 . And in real datasets, the average number of values per feature is often small, so L^2 is similar to D^2 . Approximately, the time complexity of CDE is $O(ND^2)$.

Algorithm 1 CDE (\mathcal{D} , α , β)

Input: \mathcal{D} - data set, α - proportion factor, β - dimension reducing factor
Output: \mathbf{O} - the numerical representation of objects

- 1: Generate \mathbf{M}_o and \mathbf{M}_c
- 2: Initialise $\mathbf{C}^I = \emptyset$
- 3: **for** $\mathbf{M} \in \{\mathbf{M}_o, \mathbf{M}_c\}$ **do**
- 4: Initialise $k = 2$
- 5: $rm = \emptyset$
- 6: **repeat**
- 7: $\mathbf{C}^I = [\mathbf{C}^I; kmeans(\mathbf{M}, k)]$
- 8: Remove the cluster with only one value and store the remove cluster in rm
- 9: $k+ = 1$
- 10: **until** $\frac{length(rm)}{k} \geq \alpha$
- 11: **end for**
- 12: $\mathbf{Z} = \mathbf{C}^I - mean(\mathbf{C}^I)$
- 13: Calculate the covariance matrix \mathbf{S} of \mathbf{Z}
- 14: $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{Y}] = SVD(\mathbf{S})$
- 15: $\mathbf{V} = \mathbf{Z}\mathbf{Y}^T$
- 16: Remove the columns whose range (maximum element minus minimum element) is less than β from \mathbf{V} .
- 17: Generate \mathbf{O} by the concatenation of \mathbf{V}
- 18: **return** \mathbf{O}

5.2 COSH: The CURE Instance for Outlier Detection in High-dimensional Data

In this subsection, we introduce another instance COSH for outlier detection in high-dimensional data which contains complex value interactions and is insufficiently explored. COSH use the same clustering methods, i.e., k -means, to learn multi-granularity value clusters. Different from CDE that abandons small value clusters, COSH retains them as they reflect the outlying behavior of values. Unlike CDE which uses binary cluster membership to represent the value clusters, COSH represents them with continuous dissimilarity between values and cluster centers to better quantify the outlying behavior of values. Based on the dissimilarity of value clusters, COSH learns couplings between value clusters. The object representation is the vector with outlying score of each value.

5.2.1 Learning Value Clusters for Outlier

In COSH, we also conduct k -means clustering on the two value coupling matrices. In addition to the reasons explained in Section 5.1.1, the sensitivity of k -means clustering is an important reason of using it to learn value clusters for outlier detection.

Instead of indicator matrix, we use value-cluster dissimilarity matrix to represent the clustering result for each clustering process. The definition of value-cluster dissimilarity matrix \mathbf{C}^k w.r.t. cluster number k is as follows:

$$\mathbf{C}^k = \begin{bmatrix} dis(\mathbf{v}_1, \mathbf{c}_1) & \dots & dis(\mathbf{v}_1, \mathbf{c}_k) \\ \vdots & \ddots & \vdots \\ dis(\mathbf{v}_L, \mathbf{c}_1) & \dots & dis(\mathbf{v}_L, \mathbf{c}_k) \end{bmatrix}, \quad (12)$$

where \mathbf{v} is one row of a value coupling matrix \mathbf{M} , \mathbf{c} is the centroid vector of one cluster. dis is defined as follows:

$$d(\mathbf{v}, \mathbf{c}) = \begin{cases} 0, & \text{if } \mathbf{v} \text{ and } \mathbf{c} \text{ are in different clusters} \\ \max(0, \sum_{i=1}^L \mathbf{c}(i) - \mathbf{v}(i)), & \text{otherwise.} \end{cases} \quad (13)$$

The use of the above asymmetry dissimilarity measure instead of distance measures, e.g., Euclidean distance, is decided by the semantic meaning of \mathbf{M}_o and \mathbf{M}_c . There is a basic assumption that outlying values are infrequent among all values. The value coupling matrices \mathbf{M}_o and \mathbf{M}_c are correlated with value frequency. Hence, the smaller value from \mathbf{M}_o and \mathbf{M}_c indicates the greater likelihood that it could be outlying value. Further, the value smaller than the centroid has larger chance to be an outlier than the value larger than the centroid.

5.2.2 Learning Outlying Couplings between Value Clusters

We consider two properties of outlying value and outlying value cluster: (1) The outlying value is quite different from the centroid. (2) The outlier cluster is quite different from other clusters. The value cluster matrix \mathbf{C}^k defined in Section 5.2.1 has considered the difference between a value and the centroid. We use another cluster-cluster matrix to incorporate the outlying couplings of value clusters, which is defined as follows:

$$\mathbf{D}_k = \begin{bmatrix} dis(\mathbf{c}_1, \mathbf{c}_1) & \dots & dis(\mathbf{c}_1, \mathbf{c}_k) \\ \vdots & \ddots & \vdots \\ dis(\mathbf{c}_k, \mathbf{c}_1) & \dots & dis(\mathbf{c}_k, \mathbf{c}_k) \end{bmatrix}, \quad (14)$$

where $dis(\cdot)$ is the dissimilarity defined in Equation 13.

Based on these two properties, we learn the value outlier scores w.r.t. to the value cluster difference matrix \mathbf{C}^k and cluster-cluster matrix \mathbf{D}_k as follows:

$$\mathbf{V} = \max_e \{ \mathbf{C}^{k_1} \mathbf{D}_{k_1} \mathbf{1}_{k_1}, \mathbf{C}^{k_2} \mathbf{D}_{k_2} \mathbf{1}_{k_2}, \dots \}, \quad (15)$$

where $\mathbf{1}_k$ is a vector with size k of ones, \max_e choose the element-wise maximum value across different vectors. Each entry in \mathbf{V} is the outlier score for one value. Large entry values indicate higher outlieriness.

The outlier object representation \mathbf{O} for object $x \in \mathcal{X}$ is $[\mathbf{V}_1^x, \dots, \mathbf{V}_D^x]$. The outlier score of object x is the summation of values' outlying scores, which is $outlier(x) = \sum_j^D \mathbf{V}_j^x$.

5.2.3 The COSH Method

Algorithm 2 presents the main procedures of COSH which is similar to CDE. Different from CDE, COSH represents value cluster with \mathbf{C}^i and compute the dissimilarity between value clusters in Steps (8-9); COSH uses different methods to represent values shown in Steps (14-16).

As shown in Section 5.1.3, generating \mathbf{M}_o and \mathbf{M}_c takes the complexity of $O(ND^2 + L^2)$ and clustering on the matrices has complexity $O(k_{max}L)$. Computing the outlier scores of values has complexity $O(Lk_{max}^2)$, where k_{max} is the number of times for clustering on one value matrix which is much more less than L . With the outlier scores of values, generating the outlier scores of objects has $O(ND)$. And in real datasets, the average number of values per feature is often small, so L^2 is similar to D^2 . Correspondingly, the time complexity of CDE is $O(ND^2)$.

Algorithm 2 COSH (\mathcal{D} , α)

Input: \mathcal{D} - data set, α - proportion factor
Output: \mathbf{O} - the outlier scores of all objects

- 1: Generate \mathbf{M}_o and \mathbf{M}_c
- 2: Initialize $i = 0$
- 3: **for** $\mathbf{M} \in \{\mathbf{M}_o, \mathbf{M}_c\}$ **do**
- 4: Initialize $\mathbf{k} = \emptyset$ and $j = 2$
- 5: $rm = \emptyset$
- 6: **repeat**
- 7: $\mathbf{k}(i) = j$
- 8: $\mathbf{C}^i = kmeans(\mathbf{M}, \mathbf{k}(i))$
- 9: Calculate \mathbf{D}_i
- 10: Store the clusters with one value in rm
- 11: $j+ = 1$ and $i+ = 1$
- 12: **until** $\frac{length(rm)}{k(i)} \geq \alpha$
- 13: **end for**
- 14: $\mathbf{V} = \max_e \{ \mathbf{C}^q \mathbf{D}_q \mathbf{1}_{k(q)}, q = 1, 2, \dots, i \}$,
- 15: **for each** $x \in \mathcal{X}$ **do**
- 16: $\mathbf{O}^x = [\mathbf{V}_1^x, \dots, \mathbf{V}_D^x]$
- 17: **end for**
- 18: **return** \mathbf{O}

5.3 Contrast Analysis of CDE and COSH

The CDE and COSH are both instantiated from CURE which is based on the hierarchical value couplings. The shared basis of CDE and COSH is two value coupling functions which are shown to be complementary and discriminative in Section 4.2. However, the other parts, i.e., value clusters learning and coupling learning between value clusters, are customized according to different goals of CDE and COSH. In this section, we compare these different components and analyze the intrinsic motivation of instantiation.

5.3.1 Contrastive Value Clusters

The value clusters contain abundant information so that value clusters could be customized flexibly according to different applications. In the following part, we analyze that why CDE and COSH use different value cluster learning strategies to achieve different goals.

When generating value clusters, CDE removes the small value clusters because the small value clusters have less discriminative ability for majority values and contribute less to the final clustering process. Meanwhile, COSH keeps all the small value clusters or prefers small value clusters since small clusters have higher discriminative ability for outlying values and contribute more to the outlier detection.

When representing value clusters, CDE uses the cluster membership indicator matrix \mathbf{C}^1 which keeps consensus information and differentiates values from different value clusters. Further, by multiple clustering with different cluster numbers, the value clusters group values from different granularities and keep different levels of consensus information which is helpful to distinguish similar values. Different from CDE, COSH uses the value-cluster dissimilarity matrix \mathbf{C}^k to represent value clusters which is able to differentiate two values within or across value clusters. \mathbf{C}^k keeps the most distinguishable information for each value, so that COSH could use it to give each value an outlying score and differentiate the outlier values from normal values.

5.3.2 Contrastive Value Cluster Couplings

Since CDE and COSH use different learning strategies to learn the value clusters, the couplings between value clusters are different. In the following part, we analyze why CDE and COSH learn different couplings between value clusters and use different representations.

CDE uses the concatenation of multiple cluster membership matrices to represent values, and one dimension of value representation corresponds to one value cluster. Since value clusters are generated by the same clustering methods, there are redundancy and correlations in value representation. It is better for CDE to keep all the useful discriminative information in addition to redundancy since it is designed for clustering task. Meanwhile, we expect that the dimensions of new representation are independent and uncorrelated so that the algebraic operations can be applied and for the further learning tasks. Therefore, we use PCA which does not make any information loss to eliminate the redundancy and learn linear correlation couplings in \mathbf{C}^I .

COSH is designed for outlier detection which cares the outlying behavior of values and value clusters. Accordingly, COSH uses the dissimilarity matrix \mathbf{D}_k to quantify the outlying couplings between value clusters. The value cluster which is far from other value clusters could be regarded as outlying value cluster in which the values have greater likelihood to be outlying values. Each value cluster produces one outlying score for each value which is concise and enough to distinguish the normal values and outliers. And the maximum operation across all the outlying scores from different value clusters ensure that COSH cannot miss any outlying value.

6 EXPERIMENTS FOR CLUSTERING

6.1 Experiment settings

6.1.1 Data Representation Methods and Its Parameter Settings

To test the embedding performance, CDE is compared with three popular unsupervised categorical data embedding methods: 0-1, 0-1P, and IDF. The 0-1 keeps the most complete information in the original data. The 0-1P incorporates feature correlations into the embedding. The IDF differentiates values w.r.t. frequency.

To the best of our knowledge, no existing embedding methods capture the value couplings in categorical data as in CDE. To test the CDE-based learning performance, we compute the Gaussian similarity based on CDE (denoted by CDE^G) and compare it with three typical and well-performed similarity measures which involve feature relations: COS [1], DILCA [8] and ALGO [7].

In Table 2, $|C|$ is the number of ground-truth classes in data, which is used for the clustering evaluation. We set parameter $\alpha = 10$ in CDE and parameter $\beta = 10^{-10}$ in PCA used by CDE and 0-1P. In COS, DILCA and ALGO, we use the default parameters in their original papers.

6.1.2 Data Representation Evaluation Methods

k -means clustering is used to evaluate the performance of CDE against other embedding methods. The embedding methods transform categorical data into numerical

data, hence k -means clustering can efficiently cluster objects without computing the pairwise object similarity matrix. Spectral clustering is used to evaluate the performance of this object similarity matrix against other object similarity matrices obtained by CDE^G , COS, DILCA and ALGO.

F-score and NMI [35] are two of the most popular clustering evaluation methods. Since we fix the cluster number to the number of classes in each data set, NMI performs similarly as F-score. Here we only report the results of F-score. Higher F-score indicates better clustering accuracy driven by a better embedding method or similarity measure. The p-value results are based on the paired two-tailed t-test using the null hypothesis that the clustering results of CDE and other methods come from distributions with equal means. For each data set, the F-score is the average over 50 validations of clustering with distinct starting points due to the instability of k -means clustering.

All embedding methods and similarity measures are implemented in MATLAB and clustering experiments are performed at 3.4GHz Titan Cluster with 96GB memory.

6.1.3 Data Indicators for Clustering

We use ten real-world UCI data sets from different domains for the experiments.¹ The basic data information consists of data size (denoted by $|\mathcal{X}|$), the number of features (denoted by $|\mathcal{F}|$), the number of feature values (denoted by $|\mathcal{V}|$), and the number of classes (denoted by $|C|$) for clustering are demonstrated in Table 1 and Table 2.

Various *data indicators* are used to measure the underlying characteristics of data sets, which are associated with the learning performance of embedding methods. Two key data indicators and their quantization are defined below, and the results are reported in Table 1 and Table 2.

- The *feature correlation index (FCI)* measures the average correlation strength between features:

$$FCI = \frac{2}{D(D-1)} \sum_{i=1}^{D-1} \sum_{j=i}^D SU(f_i, f_j) \quad (16)$$

SU measures the correlation between features f_i and f_j by the symmetric uncertainty [36]. Larger FCI indicates stronger correlation between features.

- The *value cluster index (VCI)* is the average of the maximum non-overlapping ratio between value sets contained in different classes for each feature:

$$VCI = \frac{1}{D} \sum_{h=1}^D \max_{i,j} \left\{ 1 - \frac{|\mathcal{V}_{C_i}^h \cap \mathcal{V}_{C_j}^h|}{|\mathcal{V}_{C_i}^h \cup \mathcal{V}_{C_j}^h|} \right\} \quad (17)$$

where $\mathcal{V}_{C_i}^h$ is the value set in class C_i for feature f_h . Larger VCI indicates the higher discriminative ability of the value sets.

6.2 Evaluation Results

CDE is firstly compared with three embedding methods, followed by a comparison with three similarity measures. We then conduct the scalability and sensitivity test of CDE.

1. <https://archive.ics.uci.edu/ml/datasets.html>

TABLE 1

F-score Results of CDE vs. Three Embedding Methods on 10 Data Sets in k -means Clustering. Note: The best performance for each data set is boldfaced. The datasets are sorted by the descending order of FCI .

| Basic Data Info. & Data Indicator | | | | F-score | | | |
|-----------------------------------|-----------------|-----------------|-------|--------------|--------------|--------------|--------------|
| Data | $ \mathcal{X} $ | $ \mathcal{Y} $ | FCI | CDE | 0-1 | 0-1P | IDF |
| Wisconsin | 683 | 89 | 0.212 | 0.967 | 0.946 | 0.946 | 0.943 |
| Soybeanssmall | 47 | 58 | 0.180 | 0.915 | 0.829 | 0.854 | 0.763 |
| Mushroom | 5644 | 97 | 0.148 | 0.731 | 0.709 | 0.694 | 0.506 |
| Mammographic | 830 | 20 | 0.116 | 0.809 | 0.793 | 0.815 | 0.517 |
| Zoo | 101 | 30 | 0.110 | 0.647 | 0.596 | 0.607 | 0.537 |
| Dermatology | 366 | 129 | 0.089 | 0.670 | 0.598 | 0.606 | 0.616 |
| Hepatitis | 155 | 36 | 0.085 | 0.680 | 0.681 | 0.667 | 0.535 |
| Adult | 30162 | 98 | 0.060 | 0.654 | 0.585 | 0.588 | 0.479 |
| Lymphography | 148 | 59 | 0.057 | 0.418 | 0.381 | 0.379 | 0.561 |
| Primarytumor | 339 | 42 | 0.020 | 0.240 | 0.230 | 0.238 | 0.190 |
| Average | | | | 0.673 | 0.635 | 0.640 | 0.565 |
| | | | | p-value | 0.003 | 0.003 | 0.020 |

6.2.1 Comparison with Three Embedding Methods

The F-scores of CDE, compared with 0-1, 0-1P and IDF, are shown in Table 1. CDE obtains the best F-score performance on seven data sets, which are significantly better than other embedding methods; and on average, it demonstrates an approximate 9%, 5% and 19% improvement over 0-1, 0-1P and IDF, respectively. The significance test results show that CDE significantly outperforms these three embedding methods at the 95% confidence level.

According to the data indicator FCI , the F-score performance of CDE, 0-1 and 0-1P has a downward trend with the decrease of FCI . CDE outperforms all the other embedding methods. This is because CDE is able to capture more sophisticated pairwise feature correlation than the other methods, which is illustrated by the performance on data sets with higher FCI , e.g., *Wisconsin*, *Soybeanssmall*, *Mammographic*, *Zoo*, *Dermatology*. This also explains the improvement of 0-1P over 0-1. Besides the couplings between features, CDE also captures the couplings across the values clusters, which makes CDE performs well on data sets with high-order feature correlation, e.g., *Adult* and *Primarytumor* that has lower FCI but may have high-order feature correlation. IDF is only sensitive to value frequency couplings, i.e., ϕ_o , while CDE is based on ϕ_o and ϕ_c which captures two complementary discriminative couplings. This explains why IDF can only obtains good results on the data sets where the objects are discriminative in terms of value frequency, e.g., *Lymphography*.

6.2.2 Comparison with Three Similarity Measures

CDE^G is compared with three well-performing feature relation-based similarity measures: COS, DILCA and ALGO. As shown in Table 2, although COS and DILCA obtain the best performance on two data sets, CDE^G remains the best performer on half of the data sets. CDE^G obtains about 8%, 3% and 5% improvement over COS, DILCA and ALGO respectively in terms of F-score. The significance test results show that CDE^G significantly outperforms the other similarity measures at the 90% confidence level. It is noted that tests on COS, DILCA and ALGO on data set *Adult* run out of memory since the computation of object similarity needs a large amount of memory. This shows that

TABLE 2

F-score Results of CDE^G vs. Three Similarity Measures on 10 Data Sets in Spectral Clustering. Note: COS, DILCA and ALGO run out of memory on *Adult*. The average values are computed according to the data sets except *Adult*.

| Basic Data Info. & Data Indicator | | | | F-score | | | |
|-----------------------------------|-----------------|-----------------|-------|--------------|--------------|--------------|-------|
| Data | $ \mathcal{F} $ | $ \mathcal{C} $ | VCI | CDE^G | COS | DILCA | ALGO |
| Wisconsin | 9 | 2 | 0.237 | 0.962 | 0.973 | 0.921 | 0.971 |
| Soybeanssmall | 21 | 4 | 0.712 | 1.000 | 0.893 | 0.910 | 0.911 |
| Mushroom | 21 | 2 | 0.310 | 0.828 | 0.825 | 0.826 | 0.826 |
| Mammographic | 4 | 2 | 0.071 | 0.817 | 0.828 | 0.826 | 0.818 |
| Zoo | 15 | 7 | 0.733 | 0.644 | 0.538 | 0.583 | 0.547 |
| Dermatology | 33 | 6 | 0.664 | 0.784 | 0.730 | 0.808 | 0.710 |
| Hepatitis | 13 | 2 | 0.141 | 0.667 | 0.463 | 0.679 | 0.662 |
| Adult | 8 | 2 | 0.032 | 0.676 | NA | NA | NA |
| Lymphography | 18 | 4 | 0.699 | 0.397 | 0.395 | 0.353 | 0.366 |
| Primarytumor | 17 | 21 | 0.873 | 0.242 | 0.196 | 0.224 | 0.209 |
| Average | | | | 0.704 | 0.649 | 0.681 | 0.669 |
| | | | | p-value | 0.050 | 0.100 | 0.032 |

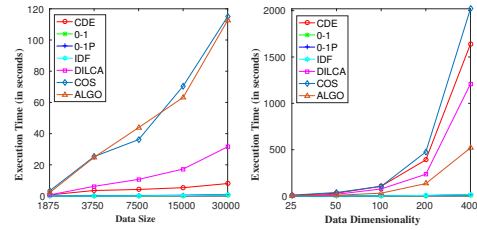


Fig. 2. Scalability Test Results.

it may be more efficient to represent categorical data with an embedding matrix than by a similarity matrix.

CDE^G achieves better performance than the other similarity measures, especially on data sets with larger VCI and larger $|\mathcal{C}|$, e.g., *Primarytumor*, *Zoo*, *Soybeanssmall*, *Lymphography*. This is because CDE^G learns the value clusters with different granularities and considers the couplings between these value clusters, which enables CDE^G to obtain more faithful value similarities than the other similarity measures that do not consider such couplings. Also, comparing to the performance of 0-1, 0-1P and IDF demonstrated in Table 1, the performance of similarity measures are better on the data sets with higher FCI , e.g., *Wisconsin*, *Soybeanssmall*, *Mushroom*, *Mammographic* according to Table 2. This is because CDE^G , COS, DILCA and ALGO are able to capture the pairwise relations between features.

6.2.3 Scalability Test

We use five subsets of the largest data set *Adult* to test the scalability w.r.t. data size. All these data subsets contain eight features. The execution time excludes the running time of clustering.

In terms of scalability w.r.t the number of features, we generate five synthetic data sets with the smallest dimension of 25 and largest dimension of 400. Each feature has two values which is randomly distributed. All the synthetic data sets have the same number of objects, i.e., 10,000.

In the left panel of Fig. 2, CDE runs significantly faster than COS, DILCA and ALGO and one order magnitude slower than 0-1, 0-1P and IDF embedding. This is because CDE is linear to the data size (N), while DILCA

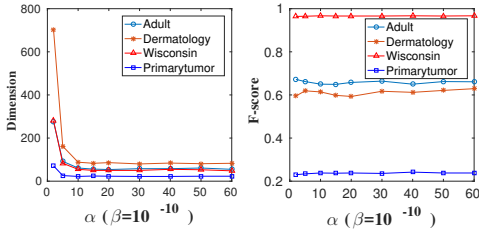


Fig. 3. Sensitivity Test of Parameter α on Four Data Sets in Terms of Dimension and F-score.

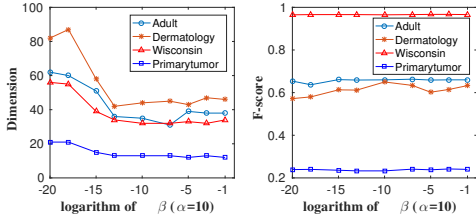


Fig. 4. Sensitivity Test of Parameter β on Four Data Sets in Terms of Dimension and F-score.

has $O(N^2 D^2 \log D)$, COS has $O(N^2 D^3 R^3)$ and ALGO has $O(N^2 D^2 + D^2 R^3)$, where R denotes the is the maximum number of distinct values for each features. In the right panel, CDE has similar runtime with COS and DILCA, and they run considerably slower than ALGO because ALGO is quadratic to the number of features (D) according to the computational complexity. All coupled methods run much slower than pure embedding methods, i.e., 0-1, 0-1Pand IDF, since modeling complex value interactions and/or feature correlations is costly.

6.2.4 Sensitivity Test

There are two parameters in CDE: α controls the dimension of value embedding before PCA and β controls the dimension of value embedding after PCA. Since the results on all data sets have the similar trend, we demonstrate the results of four data sets: *Adult*, *Dermatology*, *Wisconsin*, *Primarytumor*, which have the largest $|O|$, largest $|V|$, largest FCI and largest VCI respectively.

Fig. 3 shows the dimension of value embedding before PCA and the clustering performance with different α which directly influences the value of k in Algorithm 1. k determines the granularity of value clusters which constitute the original value embedding. Since we only drop the clusters with only one value, the clustering performance is stable with parameter α . According to Fig. 3, the dimension is stable when $\alpha \geq 10$.

Fig. 4 shows the dimension of the final value embedding and the clustering performance w.r.t. β which influences the dimension of embedding matrix during the PCA process. The smaller β value is, the higher the dimension of value embedding vector is. It shows that the performance of the clustering is stable w.r.t. β . When $\beta \geq 10^{-15}$, the dimension of value embedding vectors decreases with the increase of β on all data sets.

According to Fig. 3 and Fig. 4, the clustering performance is not sensitive to parameters α and β . The dimension is stable when $\alpha \geq 10$ and $\beta \geq 10^{-15}$.

7 EXPERIMENTS FOR OUTLIER DETECTION

7.1 Experimental Settings

7.1.1 Outlier Detectors and Their Parameter Settings

COSH represents a categorical data object with an outlying vector, so it can be applied to detect outliers directly. To evaluate the effectiveness of COSH, we compare COSH with two scoring-based representation and three other outlier detectors on 10 real-world high-dimensional data. Similar to COSH, CBRW [14] and ZERO++ (denoted by ZERO) [15] unify data representation and outlier detection as one learning task. CBRW is the state-of-the-art outlier detector for categorical data and is also a coupled method since it learns the low-level value couplings to estimate the outlier score of values. ZERO is a recently proposed subspace method for handling high-dimensional data.

The other three outlier detectors work on embedding-based representation (i.e., iForest [16]) or similarity-based representation (i.e., ABOD [37] and LOF [18]). iForest handles high-dimensional data by working on the feature subspace. ABOD is an angle-based method which is designed for high-dimensional data. LOF is one of the most popular methods which works on full dimension. To keep most complete information in original data sets and avoid introducing noise information for outlier detectors, we transform the categorical data into numerical space with one-hot encoding for enabling iForest, ABOD and LOF to work on categorical data. Another reason for using one-hot encoding instead of similarity measures is that there is no consistently effective similarity for different data sets [38] and one-hot encoding performs comparably well to other embedding- or similarity-based representation while it is much more efficient [15], [38].

COSH uses k -means, so its result is not deterministic. ZERO and iForest are also non-deterministic methods, so these three methods' performance results are taken average from 10 runs. We set parameter $\alpha = 30$ in COSH and parameter $\alpha = 0.95$ is recommended in CBRW [14]. We use $t = 50, n = 256$ in iForest and $t = 50, n = 8$ in ZERO. LOF is parameter free. Since small k is suggested in [18], we use $k = 5$ in LOF.

7.1.2 Outlier Detection Evaluation Methods

All the outlier detectors also produce a ranking based on the outlier scores. As shown in [39], the quality of ranking can be estimated by the area under ROC curve (AUC) which is computed by the Mann-Whitney-Wilcoxon test. AUC is one of the most popular performance evaluation methods and it takes the class-imbalance into consideration. Higher AUC indicates better outlier detection accuracy.

COSH is implemented in MATLAB and other five outlier detectors are implemented in JAVA. All the COSH related experiments were performed at a node 3.4GHz Titan Cluster with 96GB memory.

7.1.3 Data Sets and Data Indicators for Outlier Detection

Ten publicly available real-world data sets² are used, which cover diverse domains, e.g., Internet advertising, image

2. The used data sets are available at <http://featureselection.asu.edu>, <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/>, <http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html> and <http://tunedit.org/repo/Data>

object recognition, web page classification, and text classification. The basic data information is shown in Table 3. Six of all the data sets are directly transformed from highly imbalanced classification data, where the smallest class is treated as outliers and the largest class is regarded as normal class; and we transform the other four data sets (*PC*, *BASE*, *web*, *RELA*) by randomly sampling a small subset of the smallest class as outliers to ensure the data sets contain 2% outliers. The performance of these downsampled data sets is taken average over 10 times of sampling.

We use two data indicators to quantify the value separability and the couplings between outlier values. We define two data indicators *value separability index (VSI)* and *outlier coupling index (OCI)* below and the quantization results are shown in Table 3.

- The *value separability index (VSI)* is quantified by the value overlapping in normal objects and outlier objects which is defined as follows:

$$VSI = \min \left\{ \frac{|\{x|x \in \mathcal{X}_n \cap v_j^x \in \mathcal{V}_j^{\mathcal{X}_o}\}|}{|\mathcal{X}_n|}, j \in \mathcal{F} \right\}, \quad (18)$$

where \mathcal{X}_n is the set of normal objects and \mathcal{X}_o is the set of outlier objects, and v_j^x denotes the value of object x in feature j . The larger *VSI* indicates the weaker separability of values.

- The *outlier coupling index (OCI)* is quantified by the pointwise mutual information between outlier values and normal values, which is defined as follows:

$$OCI = \frac{pmi(v_o, v'_o)}{pmi(v_o, v'_o) + pmi(v_o, v_n)}, \quad (19)$$

where $pmi(v_o, v'_o)$ is the average pointwise mutual information within outlier values, which is calculated by $pmi(v_o, v'_o) = average\{\frac{p(v_o, v'_o)}{p(v_o)p(v'_o)}, v_o, v'_o \in \mathcal{V}_o\}$. $OCI > 0.5$ indicates that the couplings within outlier values are stronger than the couplings between outlier values and normal values.

7.2 Evaluation Results

7.2.1 Outlier Detection Effectiveness

The AUC performance of COSH and its five competitors: CBRW, ZERO, iForest, ABOD and LOF is reported in Table 3. COSH performs better than its five competitors on seven data sets, and significantly outperforms them at the 95% confidence level. On average, COSH obtains more than 17%, 27%, 39%, 29% and 44% improvement over CBRW, ZERO, iForest, ABOD and LOF, respectively. Among all outlier detection methods, COSH, CBRW and ZERO are scoring-based representation since they integrate model learning and data representation into representation, while iForest, ABOD and LOF are outlier detectors based on embedding representation. From Table 3, the performance of scoring-based representation is much better than pure outlier detectors that rely on data conversion.

In Table 3, the data sets are sorted in the descending order of *VSI*. The data indicator *VSI* describes the separability of values from single feature according to the overlapping values of outlier objects and normal objects. COSH obtains the best performance on all the data sets with higher *VSI* (e.g. $VSI > 60\%$), and it averagely achieves

TABLE 3

AUC Results of COSH vs. Five Outlier Detectors on 10 Data Sets. Note: CBRW runs out of memory on high-dimensional data *WebKB* and *Reuters8*. ABOD runs out-of-memory on large data *w7a* and *CelebA*

| Data Info. | | Data Indicator | | AUC Performance | | | | | | |
|------------|-----------------|-----------------|-------|-----------------|--------------|--------------|-------|---------|-------|--------------|
| Data | $ \mathcal{X} $ | $ \mathcal{F} $ | VSI | OCI | COSH | CBRW | ZERO | iForest | ABOD | LOF |
| w7a | 49749 | 300 | 0.950 | 0.589 | 0.835 | 0.646 | 0.538 | 0.404 | NA | 0.500 |
| CelebA | 202599 | 39 | 0.845 | 0.501 | 0.716 | 0.646 | 0.538 | 0.404 | NA | 0.500 |
| WebKB | 1658 | 6601 | 0.814 | 0.551 | 0.753 | NA | 0.698 | 0.678 | 0.670 | 0.825 |
| RELATHE | 794 | 4080 | 0.788 | 0.501 | 0.896 | 0.701 | 0.605 | 0.556 | 0.569 | 0.743 |
| BASEHOCK | 1019 | 4320 | 0.706 | 0.513 | 0.909 | 0.618 | 0.529 | 0.471 | 0.488 | 0.664 |
| PCMAC | 1002 | 3039 | 0.698 | 0.536 | 0.890 | 0.633 | 0.528 | 0.476 | 0.490 | 0.620 |
| Reuters8 | 3974 | 9467 | 0.260 | 0.552 | 0.872 | NA | 0.883 | 0.839 | 0.786 | 0.892 |
| Caltech-28 | 829 | 727 | 0.088 | 0.500 | 0.943 | 0.960 | 0.954 | 0.934 | 0.927 | 0.439 |
| Caltech-16 | 829 | 253 | 0.054 | 0.510 | 0.996 | 0.993 | 0.988 | 0.972 | 0.977 | 0.388 |
| wap.wc | 346 | 4229 | 0.038 | 0.534 | 0.975 | 0.790 | 0.657 | 0.579 | 0.524 | 0.516 |
| Average | | | | | 0.879 | 0.748 | 0.692 | 0.631 | 0.679 | 0.609 |
| | | | | | p-value | 0.023 | 0.020 | 0.002 | 0.008 | 0.010 |

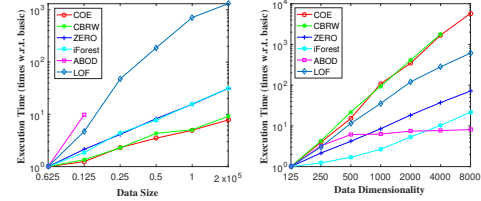


Fig. 5. Scalability Test Results. ABOD and CBRW run out of memory when the number of objects reaches 25,000 and the number of features reaches 8,000, respectively

substantial AUC improvement over its five competitors CBRW, ZERO, iForest, ABOD and LOF by more than 28%, 46%, 67%, 50% and 30%, respectively. *VSI* quantifies the separability of single feature, while some outliers could be identified by multiple features. COSH captures high-order interactions through couplings between value clusters which helps to detect outliers in data sets without strongly relevant features (i.e., low *VSI*).

OCI captures the relation between outliers and normal values across two features. The larger *OCI* is, the stronger couplings exist within outliers and weaker couplings are between outliers and normal objects. In the data sets with the highest *OCI*, i.e., *w7a*, COSH achieves much better performance than others, while COSH does not show its superiority in the data sets with the lowest *OCI*, i.e., *Cal28*.

7.2.2 Scalability Test

COSH is implemented by MATLAB while other methods are implemented by JAVA, so the absolute time is not comparable. We demonstrate the ratio of the execution time to the base time which is from the smallest data set. We use six subsets of the largest data set *CelebA* to test the scalability w.r.t. data size. All these data sets contain the same number of features, i.e., 39. The execution time on the smallest data set is: 26.6s for COSH, 0.344s for CBRW, 3.416s for ZERO, 0.299s for iForest, 3685.467s for ABOD and 2.439s for LOF.

In terms of scalability w.r.t. the number of features, seven subsets of the data sets with the largest number of features, *R8* are used. All these seven data sets contains the same number of objects, i.e., 3,974. The execution time on the smallest data set is: 88.21s for COSH, 1.657s for CBRW, 7.244s for ZERO, 0.182s for iForest, 84.345s for ABOD and 0.581s for LOF.

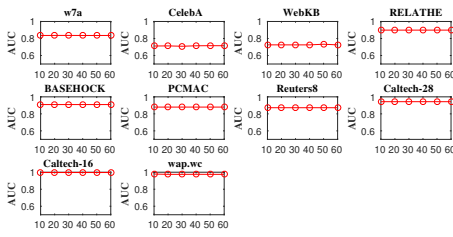


Fig. 6. Sensitivity Test Results w.r.t. α on All Ten Data Sets.

The computational complexities of CBRW, ZERO, iForest, ABOD and LOF are $O(ND^2)$, $O(ND)$, $O(ND)$, $O(N^3D)$ and $O(N^2D)$ respectively. As shown in the right panel of Fig. 5, COSH is one of the most efficient method compared with other state-of-the-art outlier detection methods w.r.t. the number of objects, since COSH is linear to the data size and quadratic to the number of features. In the left panel of Fig. 5, COSH and CBRW have similar run time and they run considerably slower than the other four detectors, since both COSH and CBRW capture complex value interactions while the other methods ignore them. Although COSH and CBRW run slower, they obtain significantly better AUC performance than their competitors, as shown in Table 3.

7.2.3 Sensitivity Test

We investigate the sensitivity test of COSH w.r.t. its only parameter α on all the 10 data sets using a wide range of α , i.e., $\{10, 20, 30, 40, 50, 60\}$. The sensitivity test results of COSH are shown in Fig. 6. COSH performs stably w.r.t. α on all data sets. The larger α means the less times of clustering and smaller number of value clusters.

8 DISCUSSIONS

CURE is a hierarchical framework which can be customized from multiple levels. We instantiate CURE by customizing the value cluster learning and coupling learning between value clusters according to different applications based on the same couplings functions. More instances may be derived by using other couplings for different applications.

The two complementary coupling functions used by CDE and COSH capture only pairwise couplings. Instantiating the CURE framework by incorporating arbitrary length patterns and their couplings may improve the discriminative ability of the low-level value coupling functions, and further improve the representation quality.

One important component of CURE is the value cluster learning, which is instantiated by k -means clustering in CDE and COSH. Although k -means has multiple advantages, it has some limitations for detecting special shape of clusters and overlapping clusters. Learning arbitrary shapes of value clusters with different clustering methods may enrich the information of value clusters. However, various kinds of value clusters may induce more heterogeneous couplings or noises. Therefore, more advanced methods may be required to capture couplings between value clusters in this case.

Another important part of CURE is the coupling learning between value clusters, which is highly related to the properties of value clusters. There may be multiple kinds of

interactions and correlations between value clusters which are also hard to interpret them directly. Incorporating more sophisticated methods to learn value couplings, e.g., neural network model, may improve the utility of each value clusters and need less assumption.

9 CONCLUSIONS

This paper proposes a novel unsupervised representation framework (CURE) for categorical data which models hierarchical value couplings in terms of feature value interactions and value cluster correlations. Following CURE, two instances CDE and COSH are respectively introduced for clustering and outlier detection, which are based on two complementary and discriminative value couplings. A contrast analysis of CDE and COSH is given to show the underlying motivation of instantiation.

Different from existing encoding-based embedding and feature correlation-based similarity measures, CDE learns the data embedding from value clusters and couplings between value clusters. Extensive experiments show that (1) CDE significantly outperforms typical embedding methods and similarity measures in clustering evaluation. (2) Two data indicators can facilitate the explanation of clustering performance on complex data sets. (3) CDE has a good scalability and is more efficient than similarity-based representation. (4) The performance of CDE is insensitive to the two parameters

Different from existing single-granularity outlier detection methods, COSH observes hierarchical outlying behaviors from values to value clusters with different granularities. Extensive experiments show that (1) COSH significantly outperforms five state-of-the-art outlier detection methods. (2) Two data indicators can facilitate the explanation of outlier detection on complex data sets. (3) COSH has a good scalability which enables it to run on high-dimensional data sets. (4) There is only one parameter in COSH and it has little influence on the outlier detection performance.

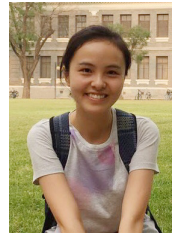
We plan to model selective value couplings to suit different applications and extend the framework into other scenarios.

ACKNOWLEDGMENTS

REFERENCES

- [1] C. Wang, X. Dong, F. Zhou, L. Cao, and C.-H. Chi, "Coupled attribute similarity learning on categorical data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 781–797, 2015.
- [2] K. Zhang, Q. Wang, Z. Chen, I. Marsic, V. Kumar, G. Jiang, and J. Zhang, "From categorical to numerical: Multiple transitive distance learning and embedding," in *Proceedings of SDM*. SIAM, 2015.
- [3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [4] L. Cao, Y. Ou, and S. Y. Philip, "Coupled behavior analysis with applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1378–1392, 2012.
- [5] A. Foss and O. R. Zaïane, "A parameterless method for efficiently discovering clusters of arbitrary shape in large datasets," in *Proceedings of ICDM*. IEEE, 2002, pp. 179–186.

- [6] A. Aizawa, "An information-theoretic perspective of tf-idf measures," *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003.
- [7] A. Ahmad and L. Dey, "A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set," *Pattern Recognition Letters*, vol. 28, no. 1, pp. 110–118, 2007.
- [8] D. Ienco, R. G. Pensa, and R. Meo, "From context to distance: Learning dissimilarity for categorical data clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, p. 1, 2012.
- [9] H. Jia, Y.-m. Cheung, and J. Liu, "A new distance metric for unsupervised learning of categorical data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 5, pp. 1065–1079, 2016.
- [10] Z. He, X. Xu, Z. J. Huang, and S. Deng, "FP-outlier: Frequent pattern based outlier detection," *Computer Science and Information Systems*, vol. 2, no. 1, pp. 103–118, 2005.
- [11] L. Akoglu, H. Tong, J. Vreeken, and C. Faloutsos, "Fast and reliable anomaly detection in categorical data," in *Proceedings of CIKM*. ACM, 2012, pp. 415–424.
- [12] M. E. Otey, A. Ghoting, and S. Parthasarathy, "Fast distributed outlier detection in mixed-attribute data sets," *Data Mining and Knowledge Discovery*, vol. 12, no. 2-3, pp. 203–228, 2006.
- [13] F. Angiulli, F. Fassetti, and L. Palopoli, "Detecting outlying properties of exceptional objects," *ACM Transactions on Database Systems*, vol. 34, no. 1, p. 7, 2009.
- [14] G. Pang, L. Cao, and L. Chen, "Outlier detection in complex categorical data by modelling the feature value couplings," in *Proceedings of IJCAI*. AAAI Press, 2016, pp. 1902–1908.
- [15] G. Pang, K. M. Ting, D. Albrecht, and H. Jin, "ZERO++: Harnessing the power of zero appearances to detect anomalies in large-scale data sets," *Journal of Artificial Intelligence Research*, vol. 57, pp. 593–620, 2016.
- [16] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, p. 3, 2012.
- [17] H.-P. Kriegel and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proceedings of SIGKDD*. ACM, 2008, pp. 444–452.
- [18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.
- [19] J. Cohen, P. Cohen, S. G. West, and L. S. Aiken, *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge, 2013.
- [20] Y. Bengio, Y. LeCun *et al.*, "Scaling learning algorithms towards ai," *Large-scale kernel machines*, vol. 34, no. 5, pp. 1–41, 2007.
- [21] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.
- [22] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, p. 391, 1990.
- [23] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.
- [24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [25] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of SIGIR*. ACM, 1999, pp. 50–57.
- [26] A. T. Wilson and P. A. Chew, "Term weighting schemes for latent dirichlet allocation," in *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, pp. 465–473.
- [27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of NIPS*, 2013, pp. 3111–3119.
- [28] C. Wang, C.-H. Chi, W. Zhou, and R. K. Wong, "Coupled interdependent attribute analysis on mixed data." in *AAAI*, 2015, pp. 1861–1867.
- [29] T. F. Cox and M. A. Cox, *Multidimensional Scaling*. CRC press, 2000.
- [30] G. E. Hinton and S. T. Roweis, "Stochastic neighbor embedding," in *Proceedings of NIPS*, 2002, pp. 833–840.
- [31] J. Song, C. Zhu, W. Zhao, W. Liu, and Q. Liu, "Model-aware representation learning for categorical data with hierarchical couplings," in *International Conference on Artificial Neural Networks*. Springer, 2017, pp. 242–249.
- [32] S. Wu and S. Wang, "Information-theoretic outlier detection for large-scale categorical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 589–602, 2013.
- [33] N. Pham and R. Pagh, "A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data," in *Proceedings of SIGKDD*. ACM, 2012, pp. 877–885.
- [34] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189–201, 2009.
- [35] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.
- [36] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proceedings of ICML*, vol. 3, 2003, pp. 856–863.
- [37] H.-P. Kriegel, A. Zimek *et al.*, "Angle-based outlier detection in high-dimensional data," in *Proceedings of SIGKDD*. ACM, 2008, pp. 444–452.
- [38] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," *red*, vol. 30, no. 2, p. 3, 2008.
- [39] D. J. Hand and R. J. Till, "A simple generalisation of the area under the roc curve for multiple class classification problems," *Machine learning*, vol. 45, no. 2, pp. 171–186, 2001.



Songlei Jian Songlei Jian is currently working toward the Ph.D. degree, jointly supervised at the National University of Defense Technology, China and the University of Technology Sydney, Australia. Her research interests include data representation, unsupervised learning and complex network analysis.



Guansong Pang Guansong Pang is a PhD candidate in the Advanced Analytics Institute at the University of Technology Sydney. Before joining AAI, he received his Master of Philosophy degree in data mining from Monash University. His research interests include data mining and non-IID learning.



Longbing Cao Longbing Cao is a Professor at the University of Technology Sydney. He has a PhD in Pattern Recognition and Intelligent Systems and another in Computing Sciences. His research interests include data science, analytics and machine learning, and behavior informatics and their enterprise applications.



Kai Lu Kai Lu is a Professor and the Deputy Dean of College of Computer Science, National University of Defense Technology, China. His research interests include parallel and distributed system software, operating systems, and big data analytics.



Hang Gao Hang Gao is a Ph.D. graduates of National University of Defense Technology, China. His research interests include machine learning, data analytics and mining.