UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Learning with Limited Labeled Data

by Yanbin Liu

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy
under the supervision of Yi Yang

University of Technology Sydney
Faculty of Engineering and Information Technology
Jan 2021

Certificate of Authorship/Originality

I, Yanbin Liu declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the 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.

Yanbin Liu

Production Note:

Signature: Signature removed prior to publication.

Date: Jan 19, 2021

ABSTRACT

Learning with Limited Labeled Data

by

Yanbin Liu

The recent success of convolutional neural networks (CNNs) relies on a large amount of annotated training data. However, many research problems suffer from the scarcity of labeled data, since annotating a large number of data is time-consuming or infeasible. This dissertation focuses on learning with limited labeled data and addresses two problems: few-shot classification and object matching.

For few-shot classification, a transductive propagation network (TPN) is first proposed to deal with the low-data issue. The idea is learning to propagate labels from labeled instances to unlabeled ones by exploiting the manifold structure of the data. Then, online feature selection with imbalanced streaming data, as a special few-shot problem, is tackled by the proposed adaptive sparse confidence-weighted (ASCW) algorithm. This algorithm utilizes the confidence-weighted (CW) learning to explore the feature correlation and maintains multiple confidence-weighted learners with different costs to address the imbalanced issue.

For object matching, since the labeled matching pairs are usually scarce, finding the potential matching among unpaired objects is important. Based on this idea, two models are proposed to solve object matching with limited labeled data. First, a squared-loss mutual information (SMI) estimator is proposed to utilize a small number of paired samples and the available unpaired ones. The estimator is formulated with optimal transport and quadratic programming in an iterative way. Second, the specific object matching problem, namely semantic correspondence, can be solved in a unified optimal transport framework. The many to one matching and background matching issues are well addressed in the proposed framework.

To evaluate the effectiveness of the aforementioned algorithms with limited labeled data, extensive experiments are conducted on various benchmark datasets, ranging from the UCI machine learning repository, few-shot image classification datasets, semantic correspondence datasets, etc.

Dissertation directed by Professor Yi Yang
AAII - Australian Artificial Intelligence Institute, UTS

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Yi Yang for the continuous support of my Ph.D. study. I first met Yi in 2014 and started pursuing Ph.D. under his supervision from 2017. Without him, I would not make up my mind to start my Ph.D. journey. During my study, I learned scientific thoughts, communication skills, and research vision from him. He also encourages me to visit companies and research institutions for collaborations, which helped me a lot during my study.

I would like to thank Linchao Zhu, who is a senior and also my friend and co-supervisor. He is enthusiastic to share ideas with me and help me revise papers. His passion and perseverance for research inspire me.

I would like to thank Juho Lee, Saehoon Kim, Sung Ju Hwang, Eunho Yang for their support during my intern at AITRICS.

I would like to thank Makoto Yamada for his help during my visiting at RIKEN AIP. His expertise, patience, and communication skills inspired me.

I would also like to thank all the members in the ReLER lab. During the study, I was fortunate to take part in many illuminating discussions with these smart people. Particularly, I would like to thank Yan Yan who helped me with my first submission.

Lastly, this thesis is dedicated to my parents for their selfless support and love.

Yanbin Liu
Sydney, Australia, 2021.

List of Publications

Conference Papers

C-1. **Liu, Y.**, Lee, J., Park, M., Kim, S., Yang, E., Hwang, S. and Yang, Y., Learning to Propagate Labels: Transductive Propagation Network for Few-shot Learning. International Conference on Learning Representations (ICLR), 2019.

C-2. **Liu, Y.**, Yan, Y., Chen, L., Han, Y. and Yang, Y., Adaptive Sparse Confidence-Weighted Learning for Online Feature Selection. AAAI Conference on Artificial Intelligence (AAAI), 2019.

C-3. **Liu, Y.**, Zhu, L., Yamada, M. and Yang, Y., Semantic Correspondence as an Optimal Transport Problem. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

Preprint Papers

P-1. **Liu, Y.**, Yamada, M., Tsai, Y.H.H., Le, T., Salakhutdinov, R. and Yang, Y., LSMI-Sinkhorn: Semi-supervised Squared-Loss Mutual Information Estimation with Optimal Transport. arXiv preprint arXiv:1909.02373, 2019.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Research Problem

Convolutional neural networks (CNNs) have achieved great success in a wide range of machine learning and computer vision problems, such as mutual information estimation [4], image classification [5, 6, 7], semantic correspondence [1, 8, 9], or semantic segmentation [10, 11]. The recent success of CNNs mainly relies on: 1) the vast computation power provided by modern GPU [12] and TPU [13], and 2) the large amount of labeled data to train the network parameters. The computation power is steadily increasing with newly designed GPU or TPU devices. However, collecting and annotating a large amount of labeled data is often laborious (e.g., pixel-wise annotation in segmentation) or even infeasible (e.g., rare-species classification), making labeled data insufficient or scare. Therefore, the scarcity of labeled data has become a bottleneck that hinders CNNs from being applied to many real-world applications, such as medical image segmentation [14], and image matching [15].

A natural way to address this limited labeled data issue is to learn transferable model using abundant external data. This idea is implemented from different perspectives, including unsupervised learning [16, 17, 18], self-supervised learning [19, 20, 21], semi-supervised learning [22], and meta-learning [23, 24, 25]. Recent unsupervised learning algorithms utilize a clustering step to generate pseudo-labels to serve as supervision signal. Clustering algorithms such as k-means [16] or Sinkhorn-Knopp [17] are employed. Self-supervised learning constructs pretext tasks from the structure or context of unlabeled data. These pretext tasks provide surrogate supervision signal for feature learning. Typical pretext tasks include jigsaw puzzle [19], rotation prediction [20], contrastive learning [21, 26, 27]. Semi-supervised learning explores the potential relationship between few labeled examples and abundant unlabeled examples. The relationship can be established by either graph-based methods such as label propagation [28, 29] (i.e., to propagate label information between labeled and unlabeled examples) or augmentation-based methods such as MixMatch [30] (i.e., to infer low-entropy labels for data-augmented unlabeled examples). The augmentation-based methods generally originates from mixup [31] and yield a series of variants such as MixMatch [30], ReMixMatch [32], FixMatch [33] and so on. Finally, meta-learning introduces a new learning paradigm: instead of sampling a mini-batch of examples from the training classes, one can create a series of learning tasks (i.e., *episodes*) with each task composed of few labeled examples called a *support set* and several unlabeled examples called a *query set*. Most meta-learning algorithms can be divided into metric-based methods that learn a transferable metric-space and optimization-based methods that learn a general updating rule with few examples.

In parallel with the aforementioned algorithms, this dissertation solves the limited labeled data issue and focuses on two specific problems: few-shot classification and object matching.

This thesis addresses few-shot classification in two different but related settings. First, for few-shot image classification, I propose *Transductive Propagation Network* (TPN), a novel meta-learning framework for transductive inference that classifies the entire test set to alleviate the low-data problem. TPN introduces the concept of *learning to propagate labels* by message passing from labeled instances to unlabeled test instances. Second, online learning with imbalanced streaming data can be seen as a special few-shot problem, since each time the algorithm receives one or few examples. I propose an adaptive sparse confidence-weighted (ASCW) algorithm to deal with this problem. This algorithm utilizes the confidence-weighted (CW) learning to explore the feature correlation and maintains multiple confidence-weighted learners with different costs to address the imbalanced issue. These two methods are highly related in the sense that both of them contain a closed-form algorithm* as the basic block. In TPN, the closed-form algorithm is label propagation [34], while in ASCW, it is confidence-weighted (CW) learning. The utilization of closed-form algorithms in few-shot classification has two advantages: (1) the closed-form algorithm is parameter-efficient and easy to solve, thus preventing the learned model from overfitting to the few-shot examples; (2) back-propagating through the closed-form algorithm can be explicitly implemented, making it easy to be incorporated into the powerful models such as deep neural networks.

For object matching, the labeled matching pairs are usually limited since the potential matching pairs of two object lists are quadratic with respect to the list size. In this situation, automatically finding high-confident matching pairs from unlabeled objects is an efficient strategy. Based on this, I propose two methods to solve object matching with limited labeled data. First, a squared-loss mutual information (SMI) estimator is proposed to utilize a small number of paired samples and the available unpaired ones. The estimator is formulated with optimal transport and quadratic programming in an iterative way. Second, a specific object matching problem, namely semantic correspondence, is solved in the unified optimal transport framework. The many to one matching and background matching issues are well addressed in the proposed framework. In both methods, the matching problem is modeled as an optimal transport problem, which is solved by the efficient Sinkhorn algorithm. Sinkhorn algorithm has a quadratic complexity and only involves simple

*Closed-form algorithm means that the problem is solved with a direct equation rather than complex optimization procedures.

matrix operations. Therefore, the back-propagation through Sinkhorn can be explicitly implemented in an parameter-efficient way, leading to natural combination with deep neural networks.

Considering both the few-shot classification and object matching problems, we summarize that the expected model for learning with limited labeled data should meet several criterions: (1) it should be parameter-efficient, which is less probable to overfit to few training examples; (2) it should induce simple and explicit gradient computation, which make it easily incorporated into powerful models such as deep neural networks. In this thesis, all the four proposed methods satisfy these criterions, either by closed-form solution or by efficient Sinkhorn iteration.

1.2 Thesis Organization

This thesis is organised as follows:

- *Chapter 2*: This chapter presents a literature survey of various research topics related to few-shot classification and object matching, including few-shot image classification, online learning, mutual information estimation, and semantic correspondence.

- *Chapter 3*: This chapter proposes a *Transductive Propagation Network* (TPN) to model transductive inference explicitly in few-shot image classification. This is realized by learning to propagate labels among data instances from unseen classes via episodic meta-learning. The contents in this chapter is published in ICLR 2019 (C-1 in the list of publications section).

- *Chapter 4*: This chapter copes with the online learning problem under imbalanced streaming data. An adaptive sparse confidence-weighted (ASCW) algorithm is proposed to model the feature correlations and alleviate the imbalanced data issue. The contents in this chapter is published in AAAI 2019 (C-2 in the list of publications section).

- *Chapter 5*: This chapter formulates the squared-loss mutual information (SMI) estimator with optimal transport and quadratic programming in an iterative way. This estimator can be applied to many object matching applications, including deep image matching and photo album summarization. The contents in this chapter is released on arxiv preprints (P-1 in the list of publications section).

- *Chapter 6*: This chapter solves the many to one matching and background matching issues in semantic correspondence. These issues are well addressed in the proposed optimal transport framework. The contents in this chapter is published in CVPR 2020 (C-3 in the list of publications section).

- *Chapter 7*: I summarize the contents and contributions of this thesis. Potential directions for future work is also given.

Among all the above Chapters, Chapter 3, 4, 5, 6 describe four main methods that show high correspondences. All four Chapters employ the parameter-efficient and computation-efficient algorithms to handle the limited labeled data in order to prevent the overfitting issue. Concretely, Chapter 3 and Chapter 4 utilize closed-form algorithms, *i.e.,* label propagation and CW. Chapter 5 and Chapter 6 make use of the efficient Sinkhorn iteration. Due to their efficiency, these algorithm can be easily combined with powerful models such as deep neural networks. More detailed descriptions are included in each Chapter.

Chapter 2

Literature Survey

Learning with limited labeled data involves many research areas, such as unsupervised learning, semi-supervised learning, self-supervised learning, and few-shot learning. This thesis focuses on two problems: few-shot classification and object matching. The two problems are closely related to few-shot learning (cf. Section 2.1), online learning (cf. Section 2.2), mutual information estimation (cf. Section 2.3), and semantic correspondence (cf. Section 2.4), which will be elaborated sequentially.

2.1 Few-shot Learning

Meta-learning. In recent works, few-shot learning often follows the idea of meta-learning [35, 36]. Meta-learning tries to optimize over batches of tasks rather than batches of data points. Each task relates to a learning problem, obtaining good performance on these tasks helps to learn quickly and generalize well to the target few-shot problem without suffering from overfitting. The well-known MAML approach [24] aims to find more transferable representations with sensitive parameters. Reptile [37] is proposed as a first-order* meta-learning approach. It is closely related to first-order MAML but does not need a training-test split for each task. Compared with the above methods, the proposed *Transductive Propagation Networks* (TPN) in this thesis has a closed-form solution for label propagation on the query points. Closed-form solution means that the problem can be solved with explicit equation form rather than complex optimization procedure. The closed-form

*First-order here means only utilize first-order derivatives rather than higher orders such as Hessian matrix.

solutions are much simpler and more efficient to compute, thus avoiding gradient computation through a complex optimization process (e.g., SGD in [24, 25, 37], SVM optimization in [38]) in the inner update[†] and usually performs faster.

Embedding and metric learning approaches. Another category of few-shot learning approach aims to optimize the transferable embedding using metric learning approaches. Matching networks [39] produce a weighted nearest neighbor classifier given the support set and adjust feature embedding according to the performance on the query set. Prototypical networks [23] first calculate the mean of all support set embedding from a given class as the class prototype. Then the transferability of feature embedding is evaluated by finding the nearest class prototype for embedded query points. An extension of prototypical networks is proposed in [28] to deal with semi-supervised few-shot learning. Relation Network [40] learns to learn a deep distance metric to compare a small number of images within episodes. The proposed *Transductive Propagation Networks* (TPN) is similar to these approaches in the sense that they all focus on learning deep embedding with strong generalization ability. However, the proposed algorithm assumes a transductive setting, in which it utilize the union of support set and query set to exploit the manifold structure of novel class space by using episodic-wise parameters.

Transduction. The setting of transductive inference was first introduced by Vapnik [41]. Transductive Support Vector Machines (TSVMs) [42] is a margin-based classification method that minimizes errors of a particular test set. It shows substantial improvements over inductive methods, especially for small training sets.

[†] Many meta-learning methods follow the inner-outer parameter update structure, which is similar to bi-level optimization. The inner-update perform adaptation to a specific task while the outer seeks for a good shared parameter initialization.

Graph-based methods [34, 43, 44, 45] are another category of transduction methods. Label propagation is proposed by [34] to transfer labels from labeled to unlabeled data instances guided by the weighted graph. Label propagation is sensitive to variance σ of the features (as shown in Eqn. 3.1), so Linear Neighborhood Propagation (LNP) [43] constructs approximated Laplacian matrix to avoid this issue. In [46], minimum spanning tree heuristic and entropy minimization are used to learn the parameter σ. In all these prior work, the graph construction is done on a pre-defined feature space using manually selected hyperparamters since it is not possible to learn them at test time. Our approach, on the other hand, is able to learn the graph construction network since it is a meta-learning framework with episodic training, where at each episode we simulate the test set with a subset of the training set.

In few-shot learning, [37] experiments with a transductive setting and shows improvements. However, they only share information between test examples via batch normalization [47] rather than explicitly model the transductive setting as in our algorithm.

2.2 Online Learning

Online learning has been extensively studied in machine learning community [48, 49, 50, 51, 52, 53]. First-order algorithms [48, 54] usually ignore the direction and scale of parameter updates. Confidence-weighted (CW) learning [49] addresses this issue by assuming a Gaussian distribution over weights with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ and solves it with closed-form solution by the KKT condition. Specifically, parameters with less confidence are updated more aggressively than more confident ones, which ensures the probability of correct classification for new instances exceeds a specified confidence. However, the aggressive update rules based on separable data assumption (*i.e.*, there exists a linear hyperplane to separate the training classes) may cause over-fitting for noisy data. Adaptive Regularization of

Weights (AROW) [51] relaxes such a separable assumption by employing a soft-margin squared hinge loss plus a confidence penalty. As another solution, Soft Confidence-weighted (SCW) [50] assigns adaptive margins for different instances.

Cost-sensitive approaches have been proposed to deal with imbalanced online learning problem, such as CSOGD [55], CSOAL [56], and MBPA [57]. They either utilize PA [48] or OGD [54] updating rules, which only considers the first-order information and ignore covariance structure. ACOG [58] adopts the idea of adaptive regularization to incorporate the second-order information, which is similar to the proposed *Adaptive Sparse Confidence-Weighted* (ASCW) algorithm in Chapter 4. However, they use ad-hoc cost values computed from the training instances while ASCW dynamically chooses the optimal cost from a set of candidates.

Many online feature selection methods have been proposed recently [59, 60, 61, 62, 57, 63, 64], most of which are first-order methods. For example, OFS [62] adopts the first-order Perceptron updating rule [65] and MBPA [57] utilizes the first-order PA [48] updating rule. Based on CW learning, [63] tries to incorporate the diagonal elements of the covariance matrix for online feature selection. However, the feature correlations are not fully explored by only using diagonal information. Compared with the above methods, the proposed ASCW algorithm not only explores feature correlations by incorporating second-order covariance structure but also selects features that can better fit the imbalanced evaluation metrics (such as F-measure, AUROC, and AUPR). Different from the regular balanced metrics such as average accuracy, the imbalanced evaluation metrics pay different attention to the examples of the majority classes and minority classes. Due to this property, our adaptive cost-selection strategy is well-devised to re-weight the majority and minority classes.

2.3 Mutual Information Estimation

The squared-loss mutual information estimator proposed in Chapter 5 is related to the general mutual information estimation, Gromov-Wasserstein, and kernelized sorting methods.

General mutual information estimation. The simplest approach to estimate the MI is estimating the probability densities $p(\boldsymbol{x}, \boldsymbol{y})$ from the paired samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, $p(\boldsymbol{x})$ from $\{\boldsymbol{x}_i\}_{i=1}^n$, and $p(\boldsymbol{y})$ from $\{\boldsymbol{y}_i\}_{i=1}^n$, respectively. However, because the estimation of the probability density is also a difficult problem, the naive approach does not tend to work well. To handle this, a density-ratio based approach is more suitable [66, 67]. More recently, a deep learning based mutual information estimation algorithm has been proposed [4]. However, these approaches still require a large number of paired samples to estimate the models. Thus, if only a limited number of paired samples are available, existing approaches are not efficient.

Most recently, the Wasserstein Dependency Measure (WDM), which measures the discrepancy between the joint probability $p(\boldsymbol{x}, \boldsymbol{y})$ and its marginals $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$, has been proposed and used for representation learning [68]. Formally, WDM is defined as follows

$$I_{\mathcal{W}}(x; y) \overset{\text{def}}{=} \mathcal{W}(p(x, y), p(x)p(y)) ,$$

where $\mathcal{W}$ is the Wasserstein distance. Since WDM can be used as an independence measure, it is highly related to the proposed LSMI-Sinkhorn algorithm in Chapter 5. The differences are that [68] focuses on finding a discriminative representation by maximizing WDM (i.e., maximize the mutual information), while LSMI-Sinkhorn focuses on estimating SMI itself.

Gromov-Wasserstein Given two set of vectors in different spaces, the Gromov-Wasserstein distance [69] can be used to find the optimal alignment between them. This method considers the pair-wise distance $D(\cdot,\cdot)$ between samples in the same set ($\{\boldsymbol{x}_i\}_{i=1}^{n_x}$ or $\{\boldsymbol{y}_j\}_{j=1}^{n_y}$) to build the distance matrix, then it finds a matching matrix $\boldsymbol{\Pi}$ ($\pi_{ij} \in \boldsymbol{\Pi}$) by minimizing the difference between the pair-wise distance matrices:

$$\begin{aligned} \min_{\boldsymbol{\Pi}} \quad & \sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\sum_{i'=1}^{n_x}\sum_{i'=1}^{n_y} \pi_{ij}\pi_{i'j'}(D(\boldsymbol{x}_i,\boldsymbol{x}_{i'}) - D(\boldsymbol{y}_j,\boldsymbol{y}_{j'}))^2, \\ \text{s.t.} \quad & \boldsymbol{\Pi}\mathbf{1}_{n_y} = \boldsymbol{a}, \boldsymbol{\Pi}^\top\mathbf{1}_{n_x} = \boldsymbol{b}, \pi_{ij} \geq 0, \end{aligned}$$

where $\boldsymbol{a} \in \Sigma_{n_x}$, $\boldsymbol{b} \in \Sigma_{n_y}$ and $\Sigma_n = \{p \in \mathbb{R}_n^+; \sum_i p_i = 1\}$ is the probability simplex. Therefore, in order to use Gromove-Wasserstein for mutual information estimation, the alignment needs to be estimated first, and then the SMI can be estimated with the aligned samples.

Computing Gromov-Wasserstein distance requires solving the quadratic assignment problem (QAP), and it is generally NP-hard for arbitrary inputs [70, 71]. In this thesis, I estimate the SMI by simultaneously solving the alignment and fitting the distribution ratio by efficiently leveraging the Sinkhorn algorithm [72] and properties of the squared-loss. I show that my approach can be considered as an example of the Gromov-Wasserstein by properly setting the cost function. Recently, semi-supervised Gromov-Wasserstein-based Optimal transport has been proposed and applied to the heterogeneous domain adaptation problems [73]. Their approach can handle tasks similar to those mentioned in this thesis. However, their method cannot be used to measure the independence.

Kernelized Sorting. The kernelized sorting [74, 75] is highly related to the Gromov-Wasserstein. Specifically, the kernelized sorting determines a set of paired samples by maximizing the Hilbert-Schmidt independence criterion (HSIC) between samples [76]. One of the constraints in kernelized sorting is to enforce the number of samples in different domains (i.e., $\{\boldsymbol{x}'_i\}_{i=1}^{n'}$ and $\{\boldsymbol{y}'_i\}_{j=1}^{n'}$) to be the same, while the proposed approach does not require such a constraint.

2.4 Semantic Correspondence

From a task perspective, the semantic correspondence problem in Chapter 6 is solved with ***hand-crafted features*** and ***CNN features***. From a methodology perspective, the proposed SCOT algorithm in Chapter 6 is related to the ***class activation map (CAM)*** and ***optimal transport*** techniques. These related works are detailed in the following.

Hand-crafted features. Early works on semantic correspondence employ hand-crafted descriptors like SIFT [77] or HOG [78] together with geometric models [79, 80]. Cho *et al.* [81] use region proposals and HOG features in Probablistic Hough Matching (PHM) algorithm for semantic matching. Ham *et al.* [82] propose a local-offset matching algorithm and introduce the PF-PASCAL benchmark.

CNN features. Recent methods employ image features from convolutional neural networks. Many of them [83, 84, 9, 85, 1, 2] are semantic flow approaches that attempts to find correspondence for individual pixel or patches. Han *et al.* [84] develop a dynamic fusion strategy based on attention mechanism to obtain a context-aware semantic representation. Lee *et al.* [9] train a CNN for semantic correspondence by using images annotated with binary foreground masks. Min *et al.* [1] use beam search algorithm on validation split of the specific dataset to find the optimal subset of deep convolutional layers.

In other methods [86, 87, 88, 3], semantic correspondence is solved as a geometric alignment problem trained with different levels of supervision. Rocco *et al.* [88] propose a two-stage regression model that utilizes self-supervision from synthetically generated images. Rocco *et al.* [3] then develop a semantic alignment model that is end-to-end trainable from weakly supervised data. Laskar *et al.* [89] cast semantic correspondence as solving a 2D point set registration problem by using keypoint-level supervision. Different from these methods, the proposed algorithm does not rely on specific kind of supervision and is flexible to use either pre-trained or finetuned models.

The comparison and visualization of several state-of-the-art semantic correspondence methods are shown in Figure 6.4. Our method only makes use of a pre-trained step on ImageNet classification task, while others either utilize a self-supervised or a weakly-supervised training step. Free from the task-specific supervision, our method is general applicable to a wider range of scenarios.

Class activation map. The idea of generating class activation map (CAM) from a classification CNN model is first introduced by Zhou *et al.* [90]. They compute a weighted sum of the feature maps of the last convolutional layer to obtain the class activation maps. Zhang *et al.* [91] then provide a simple way by directly selecting the class-specific feature maps of the last convolutional layer and prove the equivalence to [90]. Gradient-weighted Class Activation Mapping (Grad-CAM) is proposed by Selvaraju *et al.* [92]. They utilize the gradients of any target concept to produce a coarse localization map highlighting the important regions in the image for predicting the concept.

Optimal transport. Optimal transport provides a way to infer the correspondence between two distributions. Recently, it has received great attention in various computer vision tasks. Courty *et al.* [93] solve domain adaptation problem by learning a transportation plan from source domain to target domain. Su *et al.* [94] employ optimal transport to deal with the 3D shape matching and surface registration problem. Other applications include generative model [95, 96, 97, 98], graph matching [99, 100], etc. To the best of my knowledge, this is the first to model the semantic correspondence problem in the optimal transport framework.

Chapter 3

Learning to Propagate Labels: Transductive Propagation Network for Few-shot Learning

3.1 Introduction

Recent breakthroughs in deep learning [5, 6, 7] highly rely on the availability of large amounts of labeled data. However, this reliance on large data increases the burden of data collection, which hinders its potential applications to the low-data regime where the labeled data is rare and difficult to gather. On the contrary, humans have the ability to recognize new objects after observing only one or few instances [101]. For example, children can generalize the concept of "apple" after given a single instance of it. This significant gap between human and deep learning has reawakened the research interest on few-shot learning [39, 23, 24, 25, 102, 103, 104].

Few-shot learning aims to learn a classifier that generalizes well with a few examples of each of these classes. Traditional techniques such as fine-tuning [105] that work well with deep learning models would severely overfit on this task [39, 24], since a single or only a few labeled instances would not accurately represent the true data distribution and will result in the high variance classifiers. The high-variance classifiers are sensitive to the choice of training examples, which cannot generalize well to new data.

In order to solve this overfitting problem, [39] proposed a meta-learning strategy which learns over diverse classification tasks over large number of episodes rather than only on the target classification task. In each episode, the algorithm learns the embedding of the few labeled examples (the *support set*), which can be used to predict classes for the unlabeled points (the *query set*) by distance in the embedding space. The purpose of episodic training is to mimic the real test environment containing few-shot support set and unlabeled query set. The consistency between training and test environment alleviates the distribution gap and improves generalization. This episodic meta-learning strategy, due to its generalization performance, has been adapted by many follow-up work on few-shot learning. [24] learned a good initialization that can adapt quickly to the target tasks. [23] used episodes to train a good representation and predict classes by computing Euclidean distance with respect to class prototypes.

Although episodic strategy is an effective approach for few-shot learning as it aims at generalizing to unseen classification tasks, the fundamental difficulty with learning with scarce data remains for a novel classification task. One way to achieve larger improvements with limited amount of training data is to consider relationships between instances in the test set and thus predicting them as a whole, which is referred to as transduction, or transductive inference. In previous work [42, 34, 41], transductive inference has shown to outperform inductive methods which predict test examples one by one, especially in small training sets. One popular approach for transduction is to construct a network on both the labeled and unlabeled data, and propagate labels between them for joint prediction. However, the main challenge with such label propagation (and transduction) is that the label propagation network is often obtained without consideration of the main task (*i.e.*, the target evaluation task), since it is not possible to learn them at the test time.

Yet, with the meta-learning by episodic training, we can learn the label propagation network as the query examples sampled from the training set can be used to simulate the real test set for transductive inference. Motivated by this finding, we propose *Transductive Propagation Network* (TPN) to deal with the low-data prob-

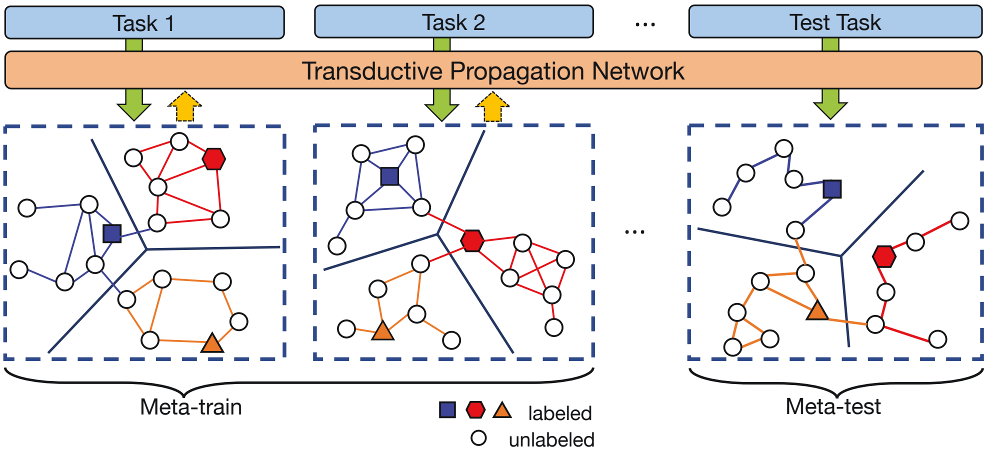


Figure 3.1 : A conceptual illustration of our transductive meta-learning framework, where lines between nodes represent graph connections and their colors represent the potential direction of label propagation. The neighborhood graph is episodic-wisely trained for transductive inference.

lem. Instead of applying the inductive inference, we utilize the entire query set for transductive inference (see Figure 3.1). Specifically, we first map the input to an embedding space using a deep neural network. Then a graph construction module is proposed to exploit the manifold structure of the novel class space using the union of support set and query set. According to the graph structure, iterative label propagation is applied to propagate labels from the support set to the query set and finally leads to a closed-form solution. With the propagated scores and ground truth labels of the query set, we compute the cross-entropy loss with respect to the feature embedding and graph construction parameters. Finally, all parameters can be updated end-to-end using backpropagation.

The main contribution of this work is threefold.

- To the best of our knowledge, we are the first to model transductive inference explicitly in few-shot learning. Although [37] experimented with a transductive setting, they only share information between test examples by batch normalization rather than directly proposing a transductive model.

- In transductive inference, we propose to *learn to propagate labels* between data instances for unseen classes via episodic meta-learning. This learned label propagation graph is shown to significantly outperform naive heuristic-based label propagation methods [34].

- We evaluate our approach on two benchmark datasets for few-shot learning, namely *mini*ImageNet and *tiered*ImageNet. The experimental results show that our *Transductive Propagation Network* outperforms the state-of-the-art methods on both datasets. Also, with semi-supervised learning, our algorithm achieves even higher performance, outperforming all semi-supervised few-shot learning baselines.

3.2 Main approach

In this section, we introduce the proposed algorithm that utilizes the manifold structure of the given few-shot classification task to improve the performance.

3.2.1 Problem definition

We follow the episodic paradigm [39] that effectively trains a meta-learner for few-shot classification tasks, which is commonly employed in various literature [23, 24, 37, 40, 106]. Given a relatively large labeled dataset with a set of classes $\mathcal{C}_{train}$, the objective of this setting is to train classifiers for an unseen set of novel classes $\mathcal{C}_{test}$, for which only a few labeled examples are available.

In each episode, a small subset of N classes are sampled from $\mathcal{C}_{train}$ to construct a *support set* and a *query set*. The *support set* contains K examples from each of the N classes (N-way K-shot setting) denoted as $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{NK}, y_{NK})\}$,

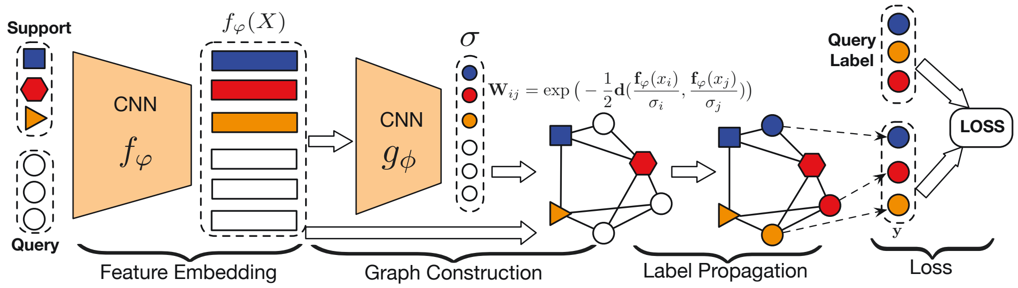

Support
Query
CNN
f_φ
$f_\varphi(X)$
CNN
g_ϕ
σ
$\mathbf{W}_{ij} = \exp\left(-\frac{1}{2}\mathbf{d}(\frac{\mathbf{f}_\varphi(x_i)}{\sigma_i}, \frac{\mathbf{f}_\varphi(x_j)}{\sigma_j})\right)$
Query Label
y
LOSS
Feature Embedding
Graph Construction
Label Propagation
Loss

Figure 3.2 : The overall framework of our algorithm in which the manifold structure of the entire query set helps to learn better decision boundary. The proposed algorithm is composed of four components: feature embedding, graph construction, label propagation, and loss generation.

while the *query set* $\mathcal{Q} = \{(\mathbf{x}_1^*, y_1^*), (\mathbf{x}_2^*, y_2^*), \ldots, (\mathbf{x}_T^*, y_T^*)\}$ includes different examples from the same N classes. Here, the support set $\mathcal{S}$ in each episode serves as the labeled training set on which the model is trained to minimize the loss of its predictions for the query set $\mathcal{Q}$. This procedure mimics training classifiers for $\mathcal{C}_{test}$ and goes episode by episode until convergence.

Meta-learning implemented by the episodic training reasonably performs well to few-shot classification tasks. Yet, due to the lack of labeled instances (K is usually very small) in the support set, we observe that a reliable classifier is still difficult to be obtained. This motivates us to consider a transductive setting that utilizes the whole query set for the prediction rather than predicting each example independently. Taking the entire query set into account, we can alleviate the low-data problem and provide more reliable generalization property.

3.2.2 Transductive Propagation Network (TPN)

We introduce *Transductive Propagation Network* (TPN) illustrated in Figure 3.2, which consists of four components: feature embedding with a convolutional neural network; graph construction that produces example-wise parameters to exploit the manifold structure; label propagation that spreads labels from the support set $\mathcal{S}$ to the query set $\mathcal{Q}$; a loss generation step that computes a cross-entropy loss between propagated labels and the ground-truths on $\mathcal{Q}$ to jointly train all parameters in the framework.

Feature embedding

We employ a convolutional neural network f_φ to extract features of an input $\mathbf{x}_i$, where $f_\varphi(\mathbf{x}_i; \varphi)$ refers to the feature map and φ indicates the parameters of the network. Despite the generality, we adopt the same architecture used in several recent works [23, 40, 39]. By doing so, we can provide more fair comparisons in the experiments, highlighting the effects of transductive approach. The network is made up of four convolutional blocks where each block begins with a 2D convolutional layer with a 3×3 kernel and filter size of 64. Each convolutional layer is followed by a batch normalization layer [47], a ReLU nonlinearity and a 2×2 max-pooling layer. We use the same embedding function f_φ for both the support set $\mathcal{S}$ and the query set $\mathcal{Q}$.

Graph construction

Manifold learning [107, 34, 108] discovers the embedded low-dimensional subspace in the data, where it is critical to choose an appropriate neighborhood graph. A common choice is Gaussian similarity function:

$$W_{ij} = \exp\left(-\frac{d(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right), \tag{3.1}$$

where $d(\cdot, \cdot)$ is a distance measure (e.g., Euclidean distance) and σ is the length scale parameter. The neighborhood structure behaves differently with respect to various σ, which means that it needs to carefully select the optimal σ for the best performance of label propagation [43, 46]. In addition, we observe that there is no principled way to tune the scale parameter in meta-learning framework, though there exist some heuristics for dimensionalty reduction methods [109, 110].

Example-wise length-scale parameter To obtain a proper neighborhood graph in meta-learning, we propose a graph construction module built on the union set of support set and query set: $\mathcal{S} \cup \mathcal{Q}$. This module is composed of a convolutional neural network g_ϕ which takes the feature map $f_\varphi(\mathbf{x}_i)$ for $\mathbf{x}_i \in \mathcal{S} \cup \mathcal{Q}$ to produce an *example-wise* length-scale parameter $\sigma_i = g_\phi(f_\varphi(\mathbf{x}_i))$. Note that the scale parameter is determined example-wisely and learned in an episodic training procedure, which adapts well to different tasks and makes it suitable for few-shot learning. With the example-wise σ_i, our similarity function is then defined as follows:

$$W_{ij} = \exp\left(-\frac{1}{2} d\Big(\frac{f_\varphi(\mathbf{x}_i)}{\sigma_i}, \frac{f_\varphi(\mathbf{x}_j)}{\sigma_j}\Big)\right) \tag{3.2}$$

where $W \in R^{(NK+T)\times(NK+T)}$ for all instances in $\mathcal{S} \cup \mathcal{Q}$. We only keep the k-max values in each row of W to construct a k-nearest neighbour graph. Then we apply the normalized graph Laplacians [107] on W, that is, $S = D^{-1/2} W D^{-1/2}$, where D is a diagonal matrix with its (i, i)-value to be the sum of the i-th row of W.

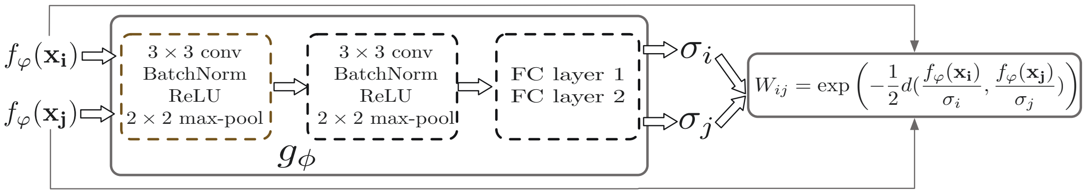

$f_\varphi(\mathbf{x_i})$
$f_\varphi(\mathbf{x_j})$
3×3 conv
BatchNorm
ReLU
2×2 max-pool
3×3 conv
BatchNorm
ReLU
2×2 max-pool
FC layer 1
FC layer 2
g_ϕ
σ_i
σ_j
$W_{ij} = \exp\left(-\frac{1}{2}d(\frac{f_\varphi(\mathbf{x_i})}{\sigma_i}, \frac{f_\varphi(\mathbf{x_j})}{\sigma_j})\right)$

Figure 3.3 : Detailed architecture of the graph construction module, in which the length-scale parameter is example-wisely determined.

Graph construction structure The structure of the proposed graph construction module is shown in Figure 3.3. It is composed of two convolutional blocks and two fully-connected layers, where each block contains a 3×3 convolution, batch normalization, ReLU activation, followed by 2×2 max pooling. The number of filters in each convolutional block is 64 and 1, respectively. To provide an example-wise scaling parameter, the activation map from the second convolutional block is transformed into a scalar by two fully-connected layers in which the number of neurons is 8 and 1, respectively.

Graph construction in each episode We follow the episodic paradigm for few-shot meta-learner training. This means that the graph is individually constructed for each task in each episode, as shown in Figure 3.1. Typically, in 5-way 5-shot training, $N = 5, K = 5, T = 75$, the dimension of W is only 100×100, which is quite efficient.

Label propagation

We now describe how to get predictions for the query set $\mathcal{Q}$ using label propagation, before the last cross-entropy loss step. Let $\mathcal{F}$ denote the set of $(NK+T) \times N$ matrix with nonnegative entries. We define a label matrix $Y \in \mathcal{F}$ with $Y_{ij} = 1$ if $\mathbf{x}_i$ is from the support set and labeled as $y_i = j$, otherwise $Y_{ij} = 0$. Starting from Y, label propagation iteratively determines the unknown labels of instances in the union set $\mathcal{S} \cup \mathcal{Q}$ according to the graph structure using the following formulation:

$$F_{t+1} = \alpha S F_t + (1-\alpha) Y \,, \tag{3.3}$$

where $F_t \in \mathcal{F}$ denotes the predicted labels at the timestamp t, S denotes the normalized weight, and $\alpha \in (0, 1)$ controls the amount of propagated information. It is well known that the sequence $\{F_t\}$ has a closed-form solution as follows:

$$F^* = (I - \alpha S)^{-1} Y \,, \tag{3.4}$$

where I is the identity matrix [34]. We directly utilize this result for the label propagation, making a whole episodic meta-learning procedure more efficient in practice.

Time complexity Matrix inversion originally takes $O(n^3)$ time complexity, which is inefficient for large n. However, in our setting, $n = NK+T$ (80 for 1-shot and 100 for 5-shot) is very small. Moreover, there is plenty of prior work on the scalability and efficiency of label propagation, such as [111, 112], which can extend our work to large-scale data.

Classification loss generation

The objective of this step is to compute the classification loss between the predictions of the union of support and query set via label propagation and the ground-truths. We compute the cross-entropy loss between predicted scores F^* and ground-truth labels from $\mathcal{S} \cup \mathcal{Q}$ to learn all parameters in an end-to-end fashion, where F^* is converted to probabilistic score using softmax:

$$P(\tilde{y}_i = j|\mathbf{x}_i) = \frac{\exp(F^*_{ij})}{\sum_{j=1}^{N} \exp(F^*_{ij})} \, . \tag{3.5}$$

Here, $\tilde{y}_i$ denotes the final predicted label for ith instance in the union of support and query set and F^*_{ij} denotes the jth component of predicted label from label propagation. Then the loss function is computed as:

$$J(\varphi, \phi) = \sum_{i=1}^{NK+T} \sum_{j=1}^{N} -\mathbb{I}(y_i == j) \log(P(\tilde{y}_i = j|\mathbf{x}_i)) \, , \tag{3.6}$$

where y_i means the ground-truth label of $\mathbf{x}_i$ and $\mathbb{I}(b)$ is an indicator function, $\mathbb{I}(b) = 1$ if b is true and 0 otherwise.

Note that in Equation (3.6), the loss is dependent on two set of parameters φ, ϕ (even though the dependency is implicit through F^*_{ij}). All these parameters are jointly updated by the episodic training in an end-to-end manner.

3.3 Experiments

We evaluate and compare our TPN with state-of-the-art approaches on two datasets, i.e., *mini*ImageNet [25] and *tiered*ImageNet [28]. The former is the most popular few-shot learning benchmark and the latter is a much larger dataset released recently for few-shot learning.

3.3.1 Datasets

***mini*ImageNet**. The *mini*ImageNet dataset is a collection of Imagenet [5] for few-shot image recognition. It is composed of 100 classes randomly selected from Imagenet with each class containing 600 examples. In order to directly compare with state-of-the-art algorithms for few-shot learning, we rely on the class splits used by [25], which includes 64 classes for training, 16 for validation, and 20 for test. All images are resized to 84 × 84 pixels.

***tiered*ImageNet**. Similar to *mini*ImageNet , *tiered*ImageNet [28] is also a subset of Imagenet [5], but it has a larger number of classes from ILSVRC-12 (608 classes rather than 100 for *mini*ImageNet). Different from *mini*ImageNet, it has a hierarchical structure of broader categories corresponding to high-level nodes in Imagenet. The top hierarchy has 34 categories, which are divided into 20 training (351 classes), 6 validation (97 classes) and 8 test (160 classes) categories. The average number of examples in each class is 1281. This high-level split strategy ensures that the training classes are distinct from the test classes semantically. This is a more challenging and realistic few-shot setting since there is no assumption that training classes should be similar to test classes. Similarly, all images are resized to 84 × 84 pixels.

3.3.2 Experimental Setup

For fair comparison with other methods, we adopt a widely-used ConvNet [24, 23] as the feature embedding function f_φ (Section 3.2.2). To push the limit of our method, we further utilize the 12-layer ResNet, following the same architecture as TADAM [113] and SNAIL [106]. The hyper-parameter k of k-nearest neighbour graph (Section 3.2.2) is set to 20 and α of label propagation is set to 0.99, as suggested in [34].

Following [23], we adopt the episodic training procedure, i.e, we sample a set of N-way K-shot training tasks to mimic the N-way K-shot test problems. Moreover, [23] proposed a "Higher Way " training strategy which used more training classes in each episode than test case. However, we find that it is beneficial to train with more examples than test phase. This is denoted as "Higher Shot" in our experiments. For 1-shot and 5-shot test problem, we adopt 5-shot and 10-shot training respectively. In all settings, the query number is set to 15 and the performance are averaged over 600 randomly generated episodes from the test set.

All our models were trained with Adam [114] and an initial learning rate of 10^{-3}. For *mini*ImageNet, we cut the learning rate in half every $10,000$ episodes and for *tiered*ImageNet, we cut the learning rate every $25,000$ episodes. The reason for larger decay step is that *tiered*ImageNet has more classes and more examples in each class which needs larger training iterations. We ran the training process until the validation loss reached a plateau.

3.3.3 Few-shot Learning Results

We compare our method with several state-of-the-art approaches in various settings. Even though the transductive method has never been used explicitly, batch normalization layer was used transductively to share information between test examples. For example, in [24, 37], they use the query batch statistics rather than global

Table 3.1 : Few-shot classification accuracies on *mini*ImageNet. All results are averaged over 600 test episodes. Top results are highlighted.

		5-way Acc		10-way Acc	
Model	Transduction	1-shot	5-shot	1-shot	5-shot
MAML [24]	BN	48.70	63.11	31.27	46.92
MAML+Transduction	Yes	50.83	66.19	31.83	48.23
Reptile [37]	No	47.07	62.74	31.10	44.66
Reptile + BN [37]	BN	49.97	65.99	32.00	47.60
PROTO NET [23]	No	46.14	65.77	32.88	49.29
PROTO NET (Higher Way) [23]	No	49.42	68.20	34.61	50.09
RELATION NET [40]	BN	51.38	67.07	34.86	47.94
Label Propagation	Yes	52.31	68.18	35.23	51.24
TPN	Yes	**53.75**	**69.43**	**36.62**	**52.32**
TPN (Higher Shot)	Yes	**55.51**	**69.86**	**38.44**	**52.77**

* "Higher Way" means using more classes in training episodes. "Higher Shot" means using more shots in training episodes. "BN" means information is shared among test examples using batch normalization.

BN parameters for the prediction, which leads to performance gain in the query set. Besides, we propose two simple transductive methods as baselines that explicitly utilize the query set. First, we propose the MAML+Transduction with slight modification of loss function to: $\mathcal{J}(\theta) = \sum_{i=1}^{T} \mathbf{y}_i \log \mathbb{P}(\widehat{\mathbf{y}}_i|\mathbf{x}_i) + \sum_{i,j=1}^{NK+T} W_{ij} \|\widehat{\mathbf{y}}_i - \widehat{\mathbf{y}}_j\|_2^2$ for transductive inference. The additional term serves as transductive regularization. Second, the naive heuristic-based label propagation methods [34] is proposed to explicitly model the transductive inference.

Experimental results on ConvNet are shown in Table 3.1 and Table3.2. Transductive batch normalization methods tend to perform better than pure inductive methods except for the "Higher Way" PROTO NET. Label propagation without

Table 3.2 : Few-shot classification accuracies on *tiered*ImageNet. All results are averaged over 600 test episodes. Top results are highlighted.

		5-way Acc		10-way Acc	
Model	Transduction	1-shot	5-shot	1-shot	5-shot
MAML [24]	BN	51.67	70.30	34.44	53.32
MAML + Transduction	Yes	53.23	70.83	34.78	54.67
Reptile [37]	No	48.97	66.47	33.67	48.04
Reptile + BN [37]	BN	52.36	71.03	35.32	51.98
PROTO NET [23]	No	48.58	69.57	37.35	57.83
PROTO NET (Higher Way) [23]	No	53.31	72.69	38.62	58.32
RELATION NET [40]	BN	54.48	71.31	36.32	58.05
Label Propagation	Yes	55.23	70.43	39.39	57.89
TPN	Yes	**57.53**	**72.85**	**40.93**	**59.17**
TPN (Higher Shot)	Yes	**59.91**	**73.30**	**44.80**	**59.44**

* "Higher Way" means using more classes in training episodes. "Higher Shot" means using more shots in training episodes. "BN" means information is shared among test examples using batch normalization.

learning to propagate outperforms other baseline methods in most cases, which verifies the necessity of transduction. The proposed TPN achieves the state-of-the-art results and surpasses all the others with a large margin even when the model is trained with regular shots. When "Higher Shot" is applied, the performance of TPN continues to improve especially for 1-shot case. This confirms that our model effectively finds the episodic-wise manifold structure of test examples through learning to construct the graph for label propagation. Experimental results on ResNet12 are shown in Table 3.3. The proposed TPN outperforms all methods except for TADAM [113] which utilizes complex architecture design and extra feature adaptation.

Table 3.3 : ResNet results on *mini*ImageNet

Method	1-shot	5-shot
SNAIL [106]	55.71	68.88
adaResNet [115]	56.88	71.94
Discriminative k-shot [116]	56.30	73.90
TADAM [113]	58.50	**76.70**
TPN	**59.46**	75.65

Another observation is that the advantages of 5-shot classification is less significant than that of 1-shot case. For example, in 5-way *mini*ImageNet , the absolute improvement of TPN over published state-of-the-art is 4.13% for 1-shot and 1.66% for 5-shot. To further investigate this, we experimented 5-way k-shot ($k = 1, 2, \cdots, 10$) experiments. The results are shown in Figure 3.4. Our TPN performs consistently better than other methods with varying shots. Moreover, it can be seen that TPN outperforms other methods with a large margin in lower shots. With the shot increase, the advantage of transduction narrows since more labelled data are used. This finding agrees with the results in TSVM [42]: when more training data are available, the bonus of transductive inference will be decreased.

Potential limitation. Taking both Table 3.3 and Figure 3.4 into account, we can see that the benefits of our method start to diminish when the model capacity or the training examples get saturated. This is reasonable since TPN is devised to deal with classification in low-data setting. Also, when training data is quite limited, we prefer to employ smaller models to avoid overfitting issues.

Large-scale applications. The experiments in this Chapter mainly focus on relatively small-scale images (*i.e.*, 84×84) for a fair comparison with the baseline methods. However, the proposed TPN and label propagation algorithm is general

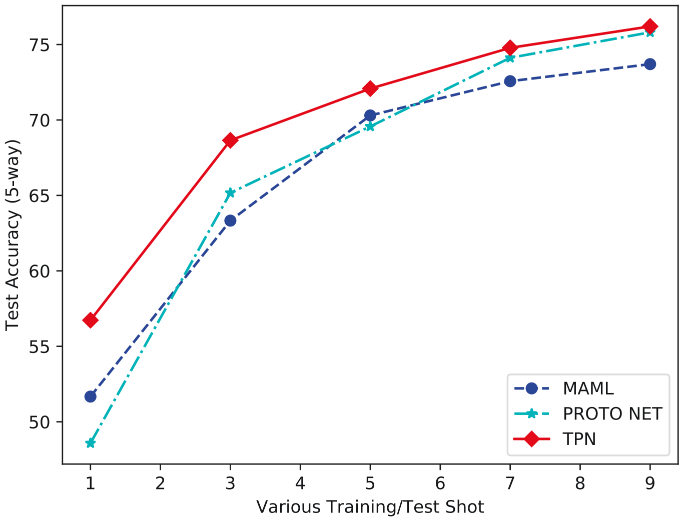


Figure 3.4 : 5-way performance with various training/test shots.

applicable to both larger networks (as shown in Table 3.3) and larger image resolutions. Recently, label propagation has been successfully combined into deep neural networks to deal with various real-world applications such as semi-supervised learning [117, 118, 119].

3.3.4 Comparison with semi-supervised few-shot learning

The main difference of traditional semi-supervised learning and transduction is the source of unlabeled data. Transductive methods directly use test set as unlabeled data while semi-supervised learning usually has an extra unlabeled set. In order to compare with semi-supervised methods, we propose a semi-supervised version of TPN, named TPN-semi, which classifies one test example each time by propagating labels from the labeled set and extra unlabeled set.

We use *mini*ImageNet and *tiered*ImageNet with the labeled/unlabeled data split proposed by [28]. Specifically, they split the images of each class into disjoint labeled and unlabeled sets. For *mini*ImageNet, the ratio of labeled/unlabeled data is 40%

Table 3.4 : Semi-supervised comparison on *mini*ImageNet.

Model	1-shot	5-shot	1-shot w/D	5-shot w/D
Soft k-Means [28]	50.09	64.59	48.70	63.55
Soft k-Means+Cluster [28]	49.03	63.08	48.86	61.27
Masked Soft k-Means [28]	50.41	64.39	49.04	62.96
TPN-semi	**52.78**	**66.42**	**50.43**	**64.95**

* “w/D” means with distraction. In this setting, many of the unlabelled data are from the so-called distraction classes, which is different from the classes of labelled data.

Table 3.5 : Semi-supervised comparison on *tiered*ImageNet.

Model	1-shot	5-shot	1-shot w/D	5-shot w/D
Soft k-Means [28]	51.52	70.25	49.88	68.32
Soft k-Means+Cluster [28]	51.85	69.42	51.36	67.56
Masked Soft k-Means [28]	52.39	69.88	51.38	69.08
TPN-semi	**55.74**	**71.01**	**53.45**	**69.93**

* “w/D” means with distraction. In this setting, many of the unlabelled data are from the so-called distraction classes, which is different from the classes of labelled data.

and 60% in each class. Likewise, the ratio is 10% and 90% for *tiered*ImageNet. All semi-supervised methods (including TPN-semi) sample support/query data from the labeled set (e.g, 40% from *mini*ImageNet) and sample unlabeled data from the unlabeled sets (e.g, 60% from *mini*ImageNet). In addition, there is a more challenging situation where many unlabelled examples from other distractor classes (different from labelled classes).

Following [28], we report the average accuracy over 10 random labeled/unlabeled splits and the uncertainty computed in standard error. Results are shown in Table 3.4 and Table 3.5. It can be seen that TPN-semi outperforms all other algorithms with a large margin, especially for 1-shot case. Although TPN is originally designed to perform transductive inference, we show that it can be successfully adapted to semi-supervised learning tasks with little modification. In certain cases where we can not get all test data, the TPN-semi can be used as an effective alternative algorithm.

3.4 Conclusion

In this chapter, we proposed the transductive setting for few-shot learning. Our proposed approach, namely *Transductive Propagation Network* (TPN), utilizes the entire test set for transductive inference. Specifically, our approach is composed of four steps: feature embedding, graph construction, label propagation, and loss computation. Graph construction is a key step that produces example-wise parameters to exploit the manifold structure in each episode. In our method, all parameters are learned end-to-end using cross-entropy loss with respect to the ground truth labels and the prediction scores in the query set. We obtained the state-of-the-art results on *mini*ImageNet and *tiered*ImageNet. Also, the semi-supervised adaptation of our algorithm achieved higher results than other semi-supervised methods. In future work, we are going to explore the episodic-wise distance metric rather than only using example-wise parameters for the Euclidean distance.

Chapter 4

Adaptive Sparse Confidence-Weighted Learning for Online Feature Selection

4.1 Introduction

Online learning typically receives and processes a single instance at a time. It has become extremely popular and been employed in many applications such as video-ad allocation [120]. In order to deal with high dimensional data streams, online feature selection (OFS) has been proposed to select a fixed number of features for prediction by an online learning fashion.

Existing online feature selection algorithms usually apply the first-order* updating rule [62, 57]. For example, OFS [62] modified the first-order Perceptron [65] algorithm by applying truncation. However, feature interactions are ignored by these algorithms. Prior studies in online learning have attested the effectiveness of second-order algorithms, such as confidence-weighted (CW) learning [49], with a covariance structure exploring the feature correlations. Due to the high computation cost of covariance matrix, very few methods [64] have been advanced for second-order online feature selection. The main implementation difference between recent first-order and second-order methods is whether to introduce a covariance structure to model the feature correlations.

While class imbalance is prevalent in real-world applications, it remains to be

*These rules are called first-order since each feature dimension is modeled independently without considering their correlations. This is in contrast to the second-order rules that takes feature correlation into account to model both the mean and covariance of the classifier weights.

under-studied in the context of online feature selection. Current online learning methods usually combine first-order updating rules with cost-sensitive[†] learning to deal with class imbalance [55, 56, 121]. In this sense, how to decide appropriate cost values is the key challenge in these methods. While most algorithms adopt fixed or ad-hoc schemes to compute costs from the given data, OMCSL [121] trains a number of classifiers with various costs and achieves improved performance.

To the best of our knowledge, no previous work has uncovered the problem of online feature selection with the presence of class imbalance. Motivated by this, we propose an Adaptive Sparse CW algorithm (ASCW) for imbalanced online-batch feature selection. Specifically, our method simultaneously maintains multiple sparse CW learners. For each learner, we assign a unique cost vector to its objective function. As the online training proceeds, we incrementally update the target measure for each learner in an online manner. For each online-batch, we choose the best performer according to their imbalanced measures in the fly for prediction. The main contributions of this chapter are summarized as follows:

- We propose an adaptive sparse CW method for feature selection on imbalanced online-batch data. Unlike previous approaches that use a fixed or ad-hoc cost vector, our method dynamically chooses the best cost from a set of candidates by incrementally updating the target performance for each learner.
- We enhance the theory of the existing sparse CW feature selection algorithm and analyze the performance behavior for F-measure.
- Empirical studies demonstrate the efficacy of the proposed algorithm. Further results show that our algorithm is capable to automatically choose a cost that

[†] Here, cost means the weights applied to different classes. Cost-sensitive means applying different weights to the majority and minority classes to ensure that the resulting classifier performs equally good on all classes.

is sufficiently close to the best one.

The remainder of the chapter is organized as follows. We first present the problem formulation and the cost-sensitive sparse CW algorithm for imbalanced feature selection. Next, we show how the adaptive strategy chooses a cost from candidates and provide theoretical analysis. We further discuss our experimental results and finally, conclude this chapter.

4.2 Imbalanced Online-Batch CW Learning

4.2.1 Notations

Table 4.1 : Notations (subindex h refers to the h-th iteration in online-batch learning).

Notations	Meaning	Notations	Meaning
$\mathbf{X}_h$	mini-batch feature	$\|\cdot\|_p$	l_p-norm
$\mathbf{y}_h \in \{-1,1\}^{N_h}$	class label	$\odot$	element-wise product
$f_h = f_h(\mathbf{X}_h)$	prediction	$[n]$	$\{1,\ldots,n\}$
$\mathbb{I}(b)$	indicator function	$\text{diag}(\cdot)$	diagonal matrix
$\boldsymbol{\mu}, \boldsymbol{\Sigma}$	mean and covariance	$\ell(\boldsymbol{\mu}; (\mathbf{x}_i, y_i))$	loss function

We first present some notations. Let superscript ^{T} represent transpose, $\mathbf{0}$ be a vector/matrix with all zeros, $\|\cdot\|_p$ denote the l_p-norm of a vector, $\text{diag}(\cdot)$ be the diagonal matrix, $A \odot B$ stand for the element-wise product of A and B, and $\mathbb{I}(b)$ be an indicator function, where $\mathbb{I}(b) = 1$ if b is true and 0 otherwise. Let $[n] = \{1,\ldots,n\}$. $\{\mathbf{X}_h, \mathbf{y}_h\}$ denote examples received at the h-th iteration, where $\mathbf{X}_h \in \mathbb{R}^{d\times N_h}$ and $\mathbf{y}_h \in \{-1,1\}^{N_h}$. $\boldsymbol{\mu}_h$ and $\boldsymbol{\Sigma}_h$ respectively represent model weights and covariance at the h-th iteration. We denote $f_h(\mathbf{X}_h) : \mathbb{R}^{d\times N_h} \to \mathbb{R}^{N_h}$ as the prediction function at the h-th iteration and $f_h = f_h(\mathbf{X}_h)$ as the predictions. The commonly-used notations are summarized in Table 4.1 for an easy understanding.

4.2.2 Cost-Sensitive Learning for Imbalanced Data

For traditional confidence-weighted learning [49, 50] and high-dimensional online feature selection such as [64], the cumulative mistake is optimized by the hinge loss as: $\ell(\boldsymbol{\mu}; (\mathbf{x}_i, y_i)) = \max(0, 1 - y_i \boldsymbol{\mu}^T \mathbf{x}_i)$. However, for imbalanced feature selection, this loss function ignores cost asymmetry between the majority classes and the minority ones. Thus, we propose the cost-sensitive loss function to deal with the imbalanced problem: $\ell_c(\boldsymbol{\mu}; (\mathbf{x}_i, y_i)) = c_+ \mathbb{I}(y_i = 1)\ell(\boldsymbol{\mu}; (\mathbf{x}_i, y_i)) + c_- \mathbb{I}(y_i = -1)\ell(\boldsymbol{\mu}; (\mathbf{x}_i, y_i))$. Let $D_i = c_+ \mathbb{I}(y_i = 1) + c_- \mathbb{I}(y_i = -1)$. Then, $\ell_c(\boldsymbol{\mu}; (\mathbf{x}_i, y_i)) = D_i \ell(\boldsymbol{\mu}; (\mathbf{x}_i, y_i))$. Moreover, we also propose $\ell_c^2(\boldsymbol{\mu}; (\mathbf{x}_i, y_i)) = D_i \ell(\boldsymbol{\mu}; (\mathbf{x}_i, y_i))^2$ as the cost-sensitive squared hinge loss.

Thus, how to choose c_+ and c_- is the key issue for imbalanced learning. We will describe the choice strategy in section 4.4, together with a theoretical analysis in detail.

4.2.3 Online-Batch CW Learning

Inspired by AROW [51] and cost-sensitive learning [55], we propose an algorithm to estimate $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ at the h-th iteration for online-batch data.

Fix $\boldsymbol{\mu}$ and update $\boldsymbol{\Sigma}$. We learn $\boldsymbol{\Sigma}$ for the following problem:

$$\min_{\boldsymbol{\Sigma}} \mathrm{D}_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| \mathcal{N}(\boldsymbol{\mu}_{h-1}, \boldsymbol{\Sigma}_{h-1})) + \frac{C}{2} \sum_{i=1}^{N_h} \mathbf{x}_i^T \boldsymbol{\Sigma} \mathbf{x}_i \,, \tag{4.1}$$

where $\mathrm{D}_{\mathrm{KL}} := \frac{1}{2}\log(\frac{\det \boldsymbol{\Sigma}_{h-1}}{\det \boldsymbol{\Sigma}}) + \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_{h-1}^{-1}\boldsymbol{\Sigma}) + \frac{1}{2}(\boldsymbol{\mu}_{h-1} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}_{h-1}^{-1}(\boldsymbol{\mu}_{h-1} - \boldsymbol{\mu}) - \frac{d}{2}$. Using KKT condition, we have

$$\boldsymbol{\Sigma}^{-1} = \boldsymbol{\Sigma}_{h-1}^{-1} + C\mathbf{X}_h \mathbf{X}_h^T \,. \tag{4.2}$$

Fix $\boldsymbol{\Sigma}$ and update $\boldsymbol{\mu}$. Once we get $\boldsymbol{\Sigma}$, we can learn $\boldsymbol{\mu}$ by the following problem:

$$\min_{\boldsymbol{\mu}} \frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu}_{h-1})^T \boldsymbol{\Sigma}_h^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}_{h-1}) + \frac{C}{q} \sum_{i=1}^{N_h} D_i \ell(\boldsymbol{\mu}; (\mathbf{x}_i, y_i))^q \,, \tag{4.3}$$

where $q = 1$ or 2.

Since $\boldsymbol{\Sigma}$ is positive semidefinite (PSD), it can be rewritten as $\boldsymbol{\Sigma} = \boldsymbol{\gamma}^2$. We introduce $\mathbf{w} := \boldsymbol{\gamma}^{-1}\boldsymbol{\mu}$, $\mathbf{w}_{h-1} := \boldsymbol{\gamma}^{-1}\boldsymbol{\mu}_{h-1}$ and $\widehat{\mathbf{x}}_i := \boldsymbol{\gamma}\mathbf{x}_i$, then problem (4.3) can be reformulated:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}_{h-1}\|_2^2 + \frac{C}{q}\sum_{i=1}^{N_h} D_i \ell(\mathbf{w}; (\widehat{\mathbf{x}}_i, y_i))^q \,. \tag{4.4}$$

In order to solve problem (4.4), we assume an online setting, i.e., each example comes sequentially from $i = 1$ to N_h. This setting is similar to PA [48]. Thus we can come up with the solution as follows:

$$\mathbf{w} = \mathbf{w}_{h-1} + \tau_i y_i \mathbf{x}_i \,, \tag{4.5}$$

$$\tau_i = \min(\ell(\mathbf{w}; (\widehat{\mathbf{x}}_i, y_i))/\|\widehat{\mathbf{x}}_i\|_2^2, CD_i) \,, \text{ for } q = 1 \,; \tag{4.6}$$

$$\tau_i = \ell(\mathbf{w}; (\widehat{\mathbf{x}}_i, y_i))/(\|\widehat{\mathbf{x}}_i\|_2^2 + 1/(2CD_i)) \,, \text{ for } q = 2 \,. \tag{4.7}$$

4.3 Sparse CW for Feature Selection

4.3.1 Feature Selection by Sparsity Index $\boldsymbol{\eta}$

The proposed online-batch CW learning algorithm maintains the full covariance matrix $\boldsymbol{\Sigma}$. It is thus not appropriate for very high-dimensional data. In practice, high-dimensional data often exhibits the property of having many zero values and only a small number of features are relevant [122]. Usually, only the relevant features and their interactions are significant for specific applications. Based on these observations, we propose the sparse feature selection algorithm in this section.

In order to find the most relevant features, we introduce an index vector $\boldsymbol{\eta} = \{0, 1\}^d$ and apply it to the feature vector $\mathbf{x}$ as $(\boldsymbol{\eta} \odot \mathbf{x})$. Here $\boldsymbol{\eta}_j = 1$ if feature j is selected and $\boldsymbol{\eta}_j = 0$ otherwise. In this situation, hinge loss is expressed as:

$$\ell(\boldsymbol{\mu}, \boldsymbol{\eta}; (\mathbf{x}_i, y_i)) = \max(0, 1 - y_i \boldsymbol{\mu}^T(\boldsymbol{\eta} \odot \mathbf{x}_i)) \,. \tag{4.8}$$

Thus $\ell_c(\boldsymbol{\mu}, \boldsymbol{\eta}; (\mathbf{x}_i, y_i)) = D_i \ell(\boldsymbol{\mu}, \boldsymbol{\eta}; (\mathbf{x}_i, y_i))$.

Considering our aim for feature selection, we impose an ℓ_0-norm constraint on $\boldsymbol{\eta}$ to induce the sparsity property, i.e., $\|\boldsymbol{\eta}\|_0 \leq r$ (where $r \ll d$). In convenience, let $\boldsymbol{\Lambda} := \{\boldsymbol{\eta} | \boldsymbol{\eta} \in \{0,1\}^d, \|\boldsymbol{\eta}\|_0 \leq r\}$ be the set of all candidate $\boldsymbol{\eta}$. So there are $|\boldsymbol{\Lambda}| = \sum_{i=0}^{r} \binom{d}{i}$ feasible $\boldsymbol{\eta}$ in total, which is exponential. In the following, we will incorporate $\boldsymbol{\eta}$ into the online-batch CW learning and solve it gradually.

At first, as in Section 4.2.3, we assume $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$ are given, and solve for $\boldsymbol{\Sigma}$. Accordingly, we incorporate $\boldsymbol{\eta}$ into equation (4.1):

$$\min_{\boldsymbol{\Sigma}} \mathrm{D}_{\mathrm{KL}}(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| \mathcal{N}(\boldsymbol{\mu}_{h-1}, \boldsymbol{\Sigma}_{h-1})) + \frac{C}{2} \sum_{i=1}^{N_h} (\boldsymbol{\eta} \odot \mathbf{x}_i)^T \boldsymbol{\Sigma} (\boldsymbol{\eta} \odot \mathbf{x_i}) \,. \tag{4.9}$$

Let $\mathbf{X}_h^r = \mathrm{diag}(\boldsymbol{\eta})\mathbf{X}_h$. Applying the KKT condition on $\boldsymbol{\Sigma}$, the closed form solution is:

$$\boldsymbol{\Sigma}(\boldsymbol{\eta})^{-1} = \boldsymbol{\Sigma}_{h-1}^{-1} + C(\mathbf{X}_h^r)(\mathbf{X}_h^r)^T \,. \tag{4.10}$$

Once we have $\boldsymbol{\Sigma}(\boldsymbol{\eta})$, we incorporate $\boldsymbol{\eta}$ into formulation (4.3) and obtain the following problem:

$$\min_{\boldsymbol{\eta} \in \boldsymbol{\Lambda}} \min_{\boldsymbol{\mu}} \frac{1}{2} (\boldsymbol{\mu} - \boldsymbol{\mu}_{h-1})^T \boldsymbol{\Sigma}_h(\boldsymbol{\eta})^{-1} (\boldsymbol{\mu} - \boldsymbol{\mu}_{h-1}) + \frac{C}{q} \sum_{i=1}^{N_h} D_i \ell(\boldsymbol{\mu}, \boldsymbol{\eta}; (\mathbf{x_i}, y_i))^q \,, \tag{4.11}$$

where $q = 1$ or 2.

Problem (4.11) is a mixed integer problem including $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$, which is hard to solve. Here, we employ the convex relaxation proposed in [123] and apply the KKT condition to transform it into the dual form as a standard convex problem:

$$\max_{\theta \in \mathbb{R}, \boldsymbol{\alpha} \in \mathcal{A}} \theta \,, \quad \text{s.t. } \theta \leq f(\boldsymbol{\alpha}, \boldsymbol{\eta}) \,, \quad \forall \boldsymbol{\eta} \in \boldsymbol{\Lambda} \,. \tag{4.12}$$

Here, $f(\boldsymbol{\alpha}, \boldsymbol{\eta})$ is defined as: $f(\boldsymbol{\alpha}, \boldsymbol{\eta}) = -\frac{1}{2} \mathbf{g}(\boldsymbol{\alpha}, \boldsymbol{\eta})^T \boldsymbol{\Sigma}_h(\boldsymbol{\eta}) \mathbf{g}(\boldsymbol{\alpha}, \boldsymbol{\eta}) - (q-1) \frac{\tilde{\boldsymbol{\alpha}}^T \tilde{\boldsymbol{\alpha}}}{2C} + \sum_{i=1}^{N_h} \alpha_i - \boldsymbol{\mu}_{h-1}^T \mathbf{g}(\boldsymbol{\alpha}, \boldsymbol{\eta})$ where $\mathbf{g}(\boldsymbol{\alpha}, \boldsymbol{\eta}) := \sum_{i=1}^{N_h} \alpha_i y_i (\boldsymbol{\eta} \odot \mathbf{x}_i)$, $\boldsymbol{\alpha} \in \mathbb{R}^{N_h}$ is the dual variable with regard to equation (4.8), $\forall i \in [N_h]$ and $\mathcal{A} := \{\boldsymbol{\alpha} \in \mathbb{R}^{N_h} | 0 \leq \alpha_i \leq U\}$ is the domain of $\boldsymbol{\alpha}$ (here, $U = CD_i$ for $q = 1$ and $U = \infty$ for $q = 2$). At last, $\tilde{\boldsymbol{\alpha}} = [\alpha_1 / D_1^{1/2}, \ldots, \alpha_{N_h} / D_{N_h}^{1/2}]$.

4.3.2 Optimization

Problem (4.12) has exponential number of constraints as $\sum_{i=0}^{r}\binom{d}{i}$, making it difficult to directly solve. Fortunately, not all constraints in (4.12) are active at optimality. Alternatively, we can efficiently solve this problem by cutting plane algorithm [124], which iteratively generate a pool of sparse feature subsets to constitute the constraints in (4.12).

Instead of considering all $T = \sum_{i=1}^{r}\binom{d}{i}$ constraints, we iteratively seek an active constraint until some stopping conditions are encountered. Given the previously estimated $\boldsymbol{\alpha}$, the most-violated constraint can be found by solving the following problem:

$$\begin{aligned}\boldsymbol{\eta}_t &= \arg\min_{\boldsymbol{\eta}\in\boldsymbol{\Lambda}} f(\boldsymbol{\alpha},\boldsymbol{\eta}) \\ &= \arg\max_{\boldsymbol{\eta}\in\boldsymbol{\Lambda}} \mathbf{g}(\boldsymbol{\alpha},\boldsymbol{\eta})^T\boldsymbol{\Sigma}_h(\boldsymbol{\eta})\mathbf{g}(\boldsymbol{\alpha},\boldsymbol{\eta}) + 2\boldsymbol{\mu}_{h-1}^T\mathbf{g}(\boldsymbol{\alpha},\boldsymbol{\eta})\,.\end{aligned} \tag{4.13}$$

Let $\mathbf{s} = \sum_{i=1}^{N_h}\alpha_i y_i \mathbf{x}_i$, then $\mathbf{g}(\boldsymbol{\alpha},\boldsymbol{\eta}) = \boldsymbol{\eta}\odot\mathbf{s}$. Problem (4.13) can be reformulated:

$$\boldsymbol{\eta}_t = \arg\max_{\boldsymbol{\eta}\in\boldsymbol{\Lambda}}(\boldsymbol{s}^T\boldsymbol{\Sigma}_{\boldsymbol{h}}(\boldsymbol{\eta}) + 2\boldsymbol{\mu}_{h-1}^T)(\boldsymbol{\eta}\odot\boldsymbol{s})\,. \tag{4.14}$$

Let $\mathbf{m} = (\boldsymbol{s}^T\boldsymbol{\Sigma}_h(\boldsymbol{\eta}) + 2\boldsymbol{\mu}_{h-1}^T)\odot\boldsymbol{s}$, then this problem can be solved by finding the r features with the largest score (e.g. m_j), and setting the corresponding η_j to 1 and the rest to 0. In other words, m_j measures the importance of the j-th feature and acts as the feature score.

After we obtained an active constraint $\boldsymbol{\eta}_t$, it can be added to the active set $\boldsymbol{\Lambda}_t = \boldsymbol{\Lambda}_{t-1}\cup\{\boldsymbol{\eta}_t\}$, then we can solve the following subproblem w.r.t constraints defined by $\boldsymbol{\Lambda}_t$:

$$\max_{\theta\in\mathbb{R},\boldsymbol{\alpha}\in\mathcal{A}} \theta, \quad \text{s.t. } \theta \le f(\boldsymbol{\alpha},\boldsymbol{\eta}), \quad \forall\boldsymbol{\eta}\in\boldsymbol{\Lambda}_t\,. \tag{4.15}$$

Problem (4.14) and (4.15) are solved alternatively and stop when: (1) $|\theta_t - \theta_{t-1}|/|\theta_t| \le \epsilon$, where ϵ is small tolerance value; (2) after $m = \lceil p/r\rceil$ iterations in order to choose p features.

4.3.3 Proximal Dual Coordinate Ascent for Subproblem (4.15)

Subproblem (4.15) regarding dual variable $\boldsymbol{\alpha}$ is hard and expensive to directly optimize. So in the following, we give a proximal-dual coordinate ascent based method to efficiently solve it. Let $K = |\boldsymbol{\Lambda}_t|$ be the number of active constraints. For each constraint $\boldsymbol{\eta}_k \in \boldsymbol{\Lambda}_t$, we take out the corresponding data, previous model parameter, model parameter and covariance matrix w.r.t $\boldsymbol{\eta}_k$ as: $\mathbf{x}_i^k \in \mathbb{R}^r$, $\boldsymbol{\mu}_k^{h-1} \in \mathbb{R}^r, \boldsymbol{\mu}_k \in \mathbb{R}^r$ and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{r\times r}$. Furthermore, let $\boldsymbol{\gamma}_k$ be the square root of $\boldsymbol{\Sigma}_k$, $\mathbf{w}_k := \boldsymbol{\gamma}_k^{-1}\boldsymbol{\mu}_k, \mathbf{w}_k^{h-1} = \boldsymbol{\gamma}_k^{-1}\boldsymbol{\mu}_k^{h-1}, \widehat{\mathbf{x}}_i^k = \boldsymbol{\gamma}_k^T\mathbf{x}_i^k$. If we denote $\mathbf{w} = [\mathbf{w}_k]_{k=1}^K, \mathbf{w}_{h-1} = [\mathbf{w}_k^{h-1}]_{k=1}^K$ and $\widehat{\mathbf{x}}_i = [\widehat{\mathbf{x}}_i^k]_{k=1}^K$. The loss function $\ell(\mathbf{w}, \boldsymbol{\eta}; (\widehat{\mathbf{x}}_i, y_i)) = \max(0, 1 - \sum_{k=1}^K y_i \mathbf{w}_k^T \widehat{\mathbf{x}}_i^k) = \max(0, 1 - y_i \mathbf{w}^T \widehat{\mathbf{x}}_i)$.

The formulation of subproblem (4.15) is formally similar to the dual format of some problems. By using KKT condition, we can obtain the primal form of subproblem (4.15):

$$\min_{\mathbf{w}} \frac{1}{2}(\sum_{k=1}^{K} \|\mathbf{w}_k - \mathbf{w}_k^{h-1}\|)^2 + \frac{C}{q}\sum_{i=1}^{N_h} D_i \ell(\mathbf{w}, \boldsymbol{\eta}; (\widehat{\mathbf{x}}_i, y_i))^q \,. \tag{4.16}$$

Problem (4.16) is non-smooth due to the $\ell_{2,1}^2$-norm regularizer. To make this problem tractable, we make some modifications and apply a proximal-dual coordinate ascent method [125] to find a nearly accurate solution of (4.16) effectively. At first, we introduce a small regularization term $\frac{\sigma}{2}\|\mathbf{w} - \mathbf{w}_{h-1}\|^2$ (i.e., $\sigma \ll 1$) and address the following optimization problem:

$$\min_{\mathbf{w}} \frac{\sigma}{2}\|\mathbf{w} - \mathbf{w}_{h-1}\|^2 + \frac{1}{2}(\sum_{k=1}^{K} \|\mathbf{w}_k - \mathbf{w}_k^{h-1}\|)^2 + \frac{C}{q}\sum_{i=1}^{N_h} D_i \ell(\mathbf{w}, \boldsymbol{\eta}; (\widehat{\mathbf{x}}_i, y_i))^q \,. \tag{4.17}$$

Remark 1. If $\mathbf{w}^*$ is an $\frac{\epsilon}{2}$-accurate minimizer of (4.17) and the σ we are choosing is sufficiently small, then $\mathbf{w}^*$ is also an ϵ-accurate solution of (4.16) [125]. Therefore, the optimal values of problems (4.16) and (4.17) are very close.

To solve problem (4.17), we then split the terms and define the following

$$\Omega(\mathbf{w}) := \frac{\sigma}{2}\|\mathbf{w} - \mathbf{w}_{h-1}\|^2 + \frac{1}{2}(\sum_{k=1}^{K} \|\mathbf{w}_k - \mathbf{w}_k^{h-1}\|)^2 \,,$$

$$L_i(\mathbf{w}^T\widehat{\mathbf{x}}_i) := \frac{1}{q}D_i\ell(\mathbf{w},\boldsymbol{\eta};(\widehat{\mathbf{x}}_i,y_i))^q\,.$$

Note here Ω is strongly convex and L_i is γ-Lipschitz for some $\gamma > 0$. Let $\Omega^*(\mathbf{z}) = \max_{\mathbf{w}} \mathbf{w}^T\mathbf{z} - \Omega(\mathbf{w})$ be the conjugate of $\Omega(\mathbf{w})$, and L_i^* be the conjugate of L_i. Then we can come up with the conjugate dual of problem (4.17):

$$\max_{\boldsymbol{\alpha}\geq\mathbf{0}} H(\boldsymbol{\alpha})\,, \tag{4.18}$$

where $H(\boldsymbol{\alpha}) = -\Omega^*(C\sum_{i=1}^{N_h}\alpha_i\widehat{\mathbf{x}}_i) - C\sum_{i=1}^{N_h}L_i^*(-\alpha_i)$.

Following [125], we define $\mathbf{z}(\boldsymbol{\alpha}) = C\sum_{i=1}^{N_h}\alpha_i\widehat{\mathbf{x}}_i$, then $\mathbf{w}(\boldsymbol{\alpha}) = \nabla^*\Omega(\mathbf{z}(\boldsymbol{\alpha}))$. Here, $\nabla^*\Omega(\mathbf{z}(\boldsymbol{\alpha}))$ denotes the gradient of the conjugate of Ω. According to the property of conjugate, it is also the solution of $\Omega^*(\mathbf{z}) = \max_{\mathbf{w}} \mathbf{w}^T\mathbf{z} - \Omega(\mathbf{w})$.[‡] Similarly, we assume an online setting as for problem (4.4). Finally, we give the full algorithm for solving the imbalanced feature selection problem in Algorithm 1.

4.3.4 Discussions

We emphasize that the proposed algorithm enhances the theory of existing sparse CW [64, 63] methods. First, with the introduction of cost-sensitive loss function in section 4.2.2, we can select features that better fit the imbalanced measures. Moreover, instead of fixing the cost, we adaptively choose the best cost from candidates and theoretically validate the optimality of our selection method in section 4.4. Second, in Equation (4.3) and (4.11), we employ $\boldsymbol{\mu}_{h-1}$ as the initialization when updating $\boldsymbol{\mu}$, while [64] uses $\mathbf{0}$. Thus $\boldsymbol{\mu}_{h-1}$ acts as the warm-start initialization and further influences on Equation (4.13) and (4.14) for solving $\boldsymbol{\eta}_t$. [64] assumes $\Sigma_h(\boldsymbol{\eta})$ to be an identity matrix when solving for $\boldsymbol{\eta}_t$. In fact, it is unclear if this assumption holds in practice. In contrast, we relax such assumption in Eq (4.13) and (4.14). Particularly, computing $\boldsymbol{\eta}_t$ reduces to a simple sorting problem in Eq (4.14). In-

[‡] This problem can be efficiently solved using Algorithm2 of [126].

Algorithm 1 Imbalanced sparse CW in online-batch manner

```
Require: Parameters $C > 0, H, r$
  Initialize $\boldsymbol{\alpha} = \frac{1}{N_h}\mathbf{1}, \boldsymbol{\mu}_0 = \mathbf{0}, \boldsymbol{\Sigma}_0 = \mathbf{I}$.
  for $h = 1 : H$ do
    Get a batch of data $\{\mathbf{X}_h, \mathbf{y}_h\}$,where $\mathbf{X}_h \in \mathbb{R}^{d \times N_h}$
    Compute $\boldsymbol{\Sigma}_h$ by (4.10) and $\boldsymbol{\gamma}$ by eigen-decomposition.
    Initialize $\Lambda_0 = \emptyset$ and $t = 1$
    while stopping conditions not meet do
      Find $\boldsymbol{\eta}_t$ by solving (4.14). Let $\Lambda_t = \Lambda_{t-1} \cup \boldsymbol{\eta}_t$.
      Compute $\mathbf{w}_k^{h-1}, \mathbf{w}_{h-1}$ according to $\Lambda_t$.
      Initialize $\mathbf{z} = \mathbf{0}, \mathbf{w} = \mathbf{w}_{h-1}$.
      for $i = 1 : N_h$ do
        Compute $D_i, \widehat{\mathbf{x}}_i^k$.
        Compute loss $\ell = \max(0, 1 - \sum_{k=1}^K y_i \mathbf{w}_k^T \widehat{\mathbf{x}}_i^k)$.
        if $\ell > 0$ then
          Compute $\alpha_i = \min(\ell/(C\|\widehat{\mathbf{x}}_i\|_2^2), D_i)$ for $q = 1$ or $\alpha_i = \ell/(C\|\widehat{\mathbf{x}}_i\|_2^2 + 0.5/D_i)$ for $q = 2$.
          Compute $\mathbf{z} = \mathbf{z} + C\alpha_i\widehat{\mathbf{x}}_i$.
          Compute $\mathbf{w} = \mathbf{w} + \nabla^*\Omega(\mathbf{z})$.
        end if
      end for
      Update $\mathbf{w}_h = \mathbf{w}, t = t + 1$.
    end while
    Update $\boldsymbol{\mu}_{\boldsymbol{h}} = \boldsymbol{\gamma}\mathbf{w}_h$.
  end for
```

tuitively, our method takes more advantages of the information from the previous online-batch through $\boldsymbol{\mu}_{h-1}$ and $\boldsymbol{\Sigma}_h(\boldsymbol{\eta})$.

4.4 Multiple Cost-Sensitive Learning

In section 4.2 and section 4.3, we propose the cost-sensitive sparse CW algorithm. However, how to decide the value of c_+ and c_- remains an issue. Some previous works use ad-hoc approaches to set up the values [55, 127, 56]. However, there is no guarantee that these approaches can achieve optimal performance for various imbalanced measures such as F-measure, AUPRC, and AUROC.

To solve this problem, we propose a strategy which maintains multiple cost-sensitive vectors. The motivation is that if multiple cost vectors $\mathbf{c} = (c_+, c_-)$ is tracked and maintained simultaneously, there must exist one setting that can best fit the data. For convenience, we assume $c_+ + c_- = 1$ to eliminate one parameter and thus $c_+ \in (0, 1)$. In order to maintain the multiple c_+, we divide $(0, 1)$ into K evenly distributed values $\theta_1, \ldots, \theta_K$, i.e., $\theta_j = j/(K+1)$ and set $c_+^j = 1 - \theta_j/2$, then the cost-sensitive loss is denoted as:

$$\ell_c^j(\boldsymbol{\mu}_j; (\mathbf{x}_i, y_i)) = (1 - \theta_j/2)\mathbb{I}(y_i = 1)\ell(\boldsymbol{\mu}_j; (\mathbf{x}_i, y_i)) + (\theta_j/2)\mathbb{I}(y_i = -1)\ell(\boldsymbol{\mu}_j; (\mathbf{x}_i, y_i))\,.$$

With this strategy, we can maintain and track K learners with the corresponding costs simultaneously: $(\theta_1, \boldsymbol{\mu}_1), \ldots, (\theta_K, \boldsymbol{\mu}_K)$. At the h-th online-batch, we update the current target measure of the j-th learner, denoted by M_h^j. Different from [121], we apply the greedy criterion to select the best performer according to $\{M_h^1, ..., M_h^K\}$ from K candidates for prediction at the h-th online-batch. With this criterion, we do not need to introduce extra hyper-parameter, and we can analyze the performance guarantee in a different way.

We update the target measures (e.g., F-measure, AUROC, and AUPRC) only using the current measure M_h^j, current predictions $\boldsymbol{f}_h$, and labels $\mathbf{y}_h$, which is efficient

Algorithm 2 Multiple Cost-Sensitive Learning.

```
Require: the number of models $K$
  Initialize $M_1^j = 0, \boldsymbol{\mu}_1^j = 0, \boldsymbol{\Sigma}_1^j = \mathbf{I}, \forall j \in [K]$.
  for $h = 1 : H$ do
    Get a batch of data $\{\mathbf{X}_h, \mathbf{y}_h\}$,where $\mathbf{X}_h \in \mathbb{R}^{d \times N_h}$
    Let $k = \arg\max_{j=1,\ldots,K} M_h^j$.
    Sample a model $\boldsymbol{\mu}_h^* = \boldsymbol{\mu}_h^k$.
    Predict for a batch of data $\mathbf{f}_h = \mathrm{sign}((\boldsymbol{\mu}_h^*)^T \mathbf{X}_h)$
    for $j = 1, \ldots, K$ do
      Update model $\boldsymbol{\mu}_h^j$ and $\boldsymbol{\Sigma}_h^j$ by running Algorithm 1.
      Compute $M_{h+1}^j$ according to $M_h^j, \boldsymbol{f}_h$, and $\mathbf{y}_h$.
    end for
  end for
```

without storing all $\boldsymbol{f}_h$ and $\mathbf{y}_h$. We summarize the multiple cost-sensitive algorithm in Algorithm 2.

4.4.1 Theoretical Analysis in F-measure

we define the following notations for binary classification:

$$\boldsymbol{a}(\theta) = [1 - \frac{\theta}{2}, \frac{\theta}{2}] \text{ and } \Delta = \frac{\theta_j - \theta_{j+1}}{2} = \frac{1}{2K},$$

$$P_1 : \text{the marginal probability of the positive instances},$$

$$\boldsymbol{E}(h) = [\texttt{fn}, \texttt{fp}] : \text{false negative and false positive},$$

$$F^* = \max_{\boldsymbol{e}} F(\boldsymbol{e}) : \text{ the maximum F-measure},$$

$$F(\boldsymbol{\mu}) : \text{ the F-measure achieved by } \boldsymbol{\mu}.$$

Proposition 1. Assume that $\{\boldsymbol{\mu}_h^1, ..., \boldsymbol{\mu}_h^K\}$ minimizes the cost-sensitive loss to a certain degree, then the F-measure achieved by Algorithm 2 has the following lower bound as long as h increases:

$$\max_{j=1,\dots,K} F(\boldsymbol{\mu}_h^j) \geq F^* - \Delta - \frac{\epsilon_0}{P_1},$$

where $k = \arg\max_{j=1,\dots,K} F(\boldsymbol{\mu}_h^j)$ and $\langle \boldsymbol{a}(\theta_k), \boldsymbol{E}(\boldsymbol{\mu}_h^k)\rangle \leq \min_{\boldsymbol{\mu}} \langle \boldsymbol{a}(\theta_k), \boldsymbol{E}(\boldsymbol{\mu})\rangle + \epsilon_0$.

Proof.

$$\begin{aligned}
& F^* - F(\boldsymbol{\mu}_h^j) \\
=& F^* - \frac{2(P_1 - \boldsymbol{E}(\boldsymbol{\mu}_h^j))}{2P_1 - \boldsymbol{E}_1(\boldsymbol{\mu}_h^j) + \boldsymbol{E}_2(\boldsymbol{\mu}_h^j)} \\
=& \frac{2P_1(F^* - 1) + \langle \boldsymbol{a}(\theta^*), \boldsymbol{E}(\boldsymbol{\mu}_h^j)\rangle}{2P_1 - \boldsymbol{E}_1(\boldsymbol{\mu}_h^j) + \boldsymbol{E}_2(\boldsymbol{\mu}_h^j)} \\
=& \frac{\langle \boldsymbol{a}(\theta^*) - \boldsymbol{a}(\theta_j), \boldsymbol{E}(\boldsymbol{\mu}_h^j)\rangle + \langle \boldsymbol{a}(\theta_j), \boldsymbol{E}(\boldsymbol{\mu}_h^j) + 2P_1(\theta^* - 1)\rangle}{2P_1 - \boldsymbol{E}_1(\boldsymbol{\mu}_h^j) + \boldsymbol{E}_2(\boldsymbol{\mu}_h^j)} \\
=& \frac{(\theta^* - \theta_j) + \langle \boldsymbol{a}(\theta_j), \boldsymbol{E}(\boldsymbol{\mu}_h^j)\rangle + 2P_1(\theta^* - 1)}{2P_1 - \boldsymbol{E}_1(\boldsymbol{\mu}_h^j) + \boldsymbol{E}_2(\boldsymbol{\mu}_h^j)} \\
\leq& \frac{(\theta^* - \theta_j) + \langle \boldsymbol{a}(\theta_j), e(\theta_j)\rangle + \epsilon_0 + 2P_1(\theta^* - 1)}{2P_1 - \boldsymbol{E}_1(\boldsymbol{\mu}_h^j) + \boldsymbol{E}_2(\boldsymbol{\mu}_h^j)} \\
=& \theta^* - \theta_j + \frac{\epsilon_0}{2P_1 - \boldsymbol{E}_1(\boldsymbol{\mu}_h^j) + \boldsymbol{E}_2(\boldsymbol{\mu}_h^j)} \\
\leq& \Delta + \frac{\epsilon_0}{P_1}.
\end{aligned} \tag{4.19}$$

□

4.5 Experiments

In this section, we evaluate the proposed ASCW algorithm on three imbalanced measures, i.e., F-measure, AUROC, and AUPRC and compare with various online learning and feature selection methods.

4.5.1 Experimental Testbed

We conduct experiments on three widely-used high-dimensional benchmarks and sample with different ratios to construct nine imbalance configurations, as shown in

Table 4.2 : Datasets Statistics

Datasets	d	N_{train}	# nonzeros per example	#Pos:#Neg
real-sim	20,958	32,309	52	1:5, 1:10, 1:20
rcv1	47,236	20,242	74	1:5, 1:10, 1:20
news20	62,061	15,935	80	1:5, 1:10, 1:19

Table 4.2. In order to construct imbalanced configurations from the original datasets, we adopt two strategies. Firstly, for binary datasets (real-sim and rcv1), we fix the negative class and sample from the positive class to satisfy specific ratios (1:5, 1:10 and 1:20). Secondly, for the multi-class dataset (news20), we set class1 as positive class and select class2-6, class2-11 and class2-20 as negative class respectively.

4.5.2 Comparison Algorithms

We compare the following algorithms:

- OFS [62]: The state-of-the-art first-order online feature selection via sparse projection.
- MBPA [57]: Margin-based passive aggressive method for online feature selection.
- CSOAL [56]: A cost-sensitive online active learning method.
- SBCW1 and SBCW2 [64]: Two variations of the sparse online-batch feature selection method.
- FGM [128]: The full-batch high-dimensional feature selection method which generates a pool of violated sparse feature subsets and combines them via efficient Multiple Kernel Learning (MKL) algorithm.

- L1SVM [129]: ℓ_1-norm SVM by Liblinear.
- ASCW1 and ASCW2: The proposed algorithm with hinge ($q = 1$) and squared hinge ($q = 2$) loss.

SBCW and ASCW consider second-order structure while others only optimize first-order information.

4.5.3 Experimental Results

As shown in Table 4.2, the number of nonzeros per example varies from 52 to 80 in different datasets, so in the experiments we set the selected feature dimension to 50 for all algorithms except that for CSOAL we set query ratio to be 1%.[§] Following [130], we repeat all online learning experiments 20 times with random permutation of training data. For full batch methods (FGM, L1SVM), we follow the default settings.

Batch Size. In Algorithm 1, $\boldsymbol{\mu}$ is updated in a pure online manner and $\boldsymbol{\Sigma}$ is updated in an online-batch manner. To explain the necessity of the online-batch update and explore proper batch size, we perform experiments on news20 with various batch sizes, as shown in Table 4.3. The best performance is achieved with batch size=1 (the strict online case). However, the time cost is unbearable. The performance of batch size=256 is close to that of 64, but 256 is 3∼4 times faster. We thus set batch size=256 in remaining experiments.

Online Performance. To compare the online performances, we evaluate three measures on all datasets. The results of ratio 1:10 are shown in Figure 4.1. It can be seen that ASCW outperforms all other methods when the number of samples

[§] Actually, 1% of all examples contain more information than 50 features of entire features for all datasets.

Table 4.3 : Test performance on news20 with various batch sizes

Ratio	Batch	news20			
		F	AUROC	AUPRC	Time (s)
1:5	256	0.5614	0.8474	0.6011	0.71
	64	0.5640	0.8536	0.6163	2.40
	1	0.9125	0.9922	0.9769	913.86
1:10	256	0.4239	0.8114	0.4306	1.39
	64	0.4275	0.8096	0.4309	4.53
	1	0.8549	0.9872	0.9479	3034.95
1:19	256	0.3349	0.8223	0.3126	2.86
	64	0.3031	0.7952	0.2867	9.63
	1	0.8080	0.9796	0.9068	13950.78

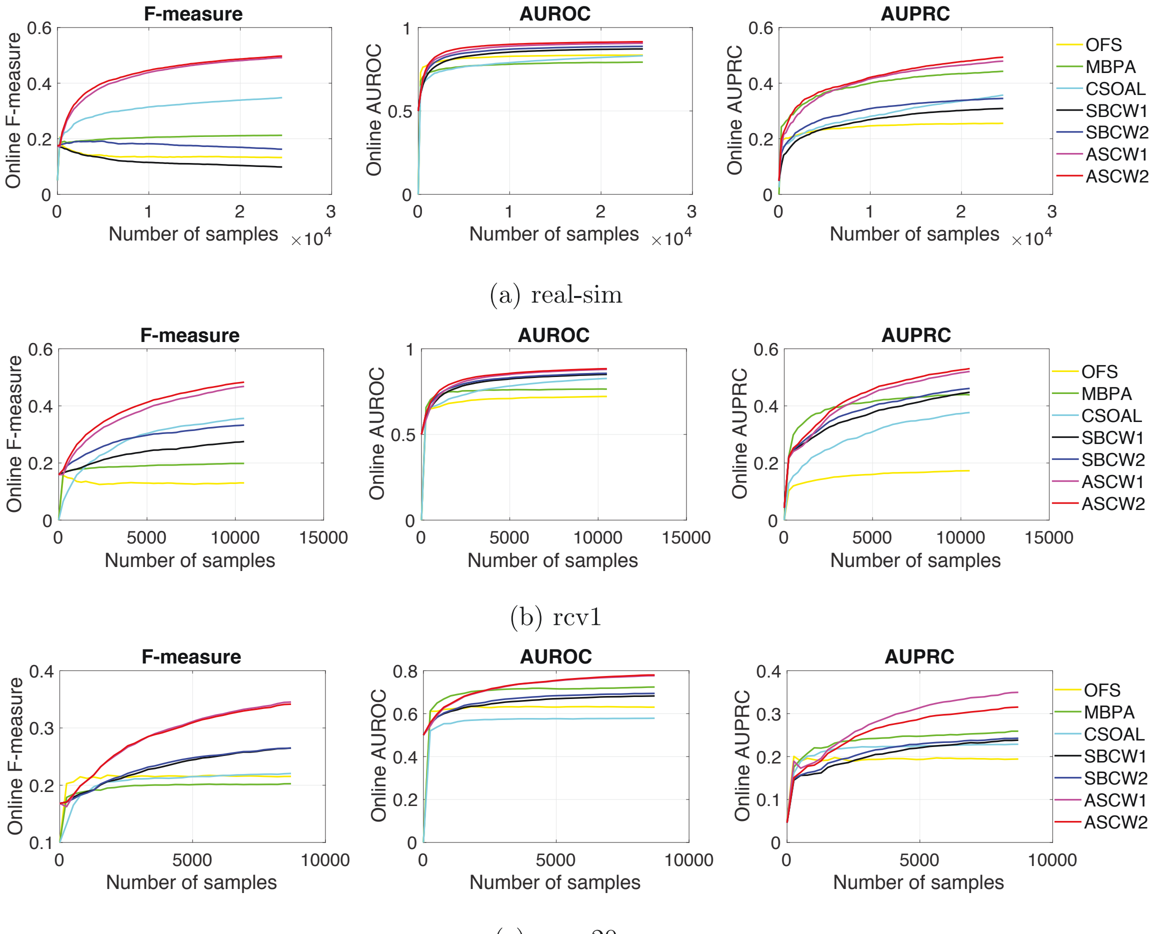
increases. Moreover, the F-measure of ASCW outperforms all other methods with a large margin.

Test Performance. We report the test performances of all algorithms under different imbalance ratios in Table 4.4. It is observed that ASCW outperforms all other algorithms on most settings for the three performance measures. Also, the improvements of ASCW on F-measure are higher than that on AUROC and AUPRC.

We attribute the good online and test performance of ASCW to two main reasons. First, our algorithm is capable of selecting a close-to-optimal cost vector $[c_+, c_-]$, which makes it perform better on imbalanced measures. Moreover, there is a theoretical guarantee on the lower bound of F-measure. It explains the higher improvements of F-measure compared with AURROC and AUPRC. Second, our algorithm employs covariance structure that can better capture the interplays among features to find more effective features.

Table 4.4 : Average test performance over models trained on 20 random data permutations.

Ratio	Methods	real-sim			rcv1			news20		
		F-measure	AUROC	AUPRC	F-measure	AUROC	AUPRC	F-measure	AUROC	AUPRC
1:5	OFS	0.0279	0.8943	0.5707	0.0215	0.8207	0.5212	0.3744	0.7706	0.4919
	MBPA	0.3742	0.8435	0.6886	0.3404	0.8365	0.6646	0.3193	0.7239	0.3930
	CSOAL	0.6248	0.9163	0.6868	**0.6366**	0.9183	0.7203	0.3579	0.5915	0.3597
	FGM	0.5334	0.9103	0.7366	0.4115	0.7235	0.4381	0.2754	0.5930	0.2411
	L1SVM	0.4501	0.9127	0.6892	0.5552	0.8906	0.7308	0.4135	0.8028	0.5664
	SBCW1	0.3778	0.9295	0.7161	0.4156	0.8913	0.7083	0.4115	0.7934	0.5298
	SBCW2	0.4363	0.9390	0.7357	0.4078	0.9056	0.7255	0.4703	0.8122	0.5474
	ASCW1	**0.7036**	**0.9434**	**0.7395**	**0.6409**	**0.9185**	**0.7398**	**0.5614**	**0.8474**	**0.6011**
	ASCW2	**0.6948**	**0.9464**	**0.7521**	0.6355	**0.9312**	**0.7887**	**0.5698**	**0.8529**	**0.6095**
1:10	OFS	0.0021	0.8537	0.2964	0.0001	0.7794	0.2480	0.2237	0.7139	0.2928
	MBPA	0.2394	0.8203	0.5564	0.2085	0.7505	0.4509	0.2096	0.7203	0.2618
	CSOAL	0.3931	0.8801	0.4528	0.4342	0.8869	0.5051	0.2526	0.6013	0.2567
	FGM	0.3580	0.9071	**0.6279**	0.2968	0.7565	0.3432	0.1942	0.6078	0.1450
	L1SVM	0.0816	0.8655	0.3473	0.2830	0.8485	0.4968	0.3986	0.7636	0.4056
	SBCW1	0.0710	0.8994	0.3778	0.3510	0.8767	0.5557	0.3193	0.7198	0.2895
	SBCW2	0.1241	0.9173	0.4273	0.3906	0.8990	**0.6016**	0.3063	0.7235	0.2821
	ASCW1	**0.5621**	**0.9422**	0.5914	**0.5649**	**0.9060**	0.5880	**0.4239**	**0.8114**	**0.4306**
	ASCW2	**0.5662**	**0.9457**	**0.6073**	**0.6005**	**0.9190**	**0.6252**	**0.4312**	**0.8188**	**0.4341**
1:20 (1:19)	OFS	0.0041	0.8295	0.1559	0.0000	0.7623	0.1191	0.1525	0.7160	0.1969
	MBPA	0.1144	0.7578	0.3941	0.1231	0.7129	0.3170	0.1283	0.7419	0.2017
	CSOAL	0.2128	0.8156	0.1776	0.2386	0.8512	0.2856	0.2570	0.6519	0.2358
	FGM	0.1827	0.8660	0.4540	0.1761	0.7224	0.2055	0.1622	0.6981	0.1282
	L1SVM	0.0000	0.8223	0.1264	0.0000	0.8310	0.2264	0.3154	0.7499	**0.2991**
	SBCW1	0.0554	0.8678	0.1947	0.2422	0.8508	0.3668	0.2538	0.7328	0.2161
	SBCW2	0.0802	0.9069	0.2466	0.2857	0.8726	0.4404	0.2455	0.7427	0.1964
	ASCW1	**0.3893**	**0.9178**	**0.4803**	**0.4565**	**0.9042**	**0.5189**	**0.3349**	**0.8223**	**0.3126**
	ASCW2	**0.4236**	**0.9372**	**0.4582**	**0.5081**	**0.9078**	**0.5073**	**0.3245**	**0.8173**	0.2937



(a) real-sim

(b) rcv1

(c) news20

Figure 4.1 : Online performance with imbalance ration 1:10 for different performance measures

4.5.4 Optimal Cost Vector

In preposition 1, we theoretically analyze the lower-bound of the F-measure achieved by Algorithm 2. In order to quantitatively verify that our algorithm can choose near to optimal cost vector $[c_+, c_-]$, we perform cost-sensitive feature selection by Algorithm 1 with costs vary among $c_+ = \{0.55, 0.60, \ldots, 0.95\}$ and choose the best cost according to overall online performance, denoted by c_+^*. To compare our selected cost with the best performance cost, we average c_+ sampled in Algorithm 2 in the last 20 iterations (>5000 examples) as an estimation of the best cost, denoted

Table 4.5 : Average estimated error of cost $\hat{c}_+$ by Algorithm 2 and optimal cost c_+^*.

Ratio	Methods	real-sim			rcv1			news20		
		F-measure	AUROC	AUPRC	F-measure	AUROC	AUPRC	F-measure	AUROC	AUPRC
1:5	ASCW1	0.0008	0.0000	0.0000	0.0090	0.0032	0.0021	0.0073	0.0058	0.0140
	ASCW2	0.0056	0.0031	0.0078	0.0082	0.0208	0.0345	0.0198	0.0020	0.0103
1:10	ASCW1	0.0000	0.0006	0.0014	0.0014	0.0084	0.0201	0.0006	0.0017	0.0119
	ASCW2	0.0071	0.0006	0.0040	0.0111	0.0235	0.0117	0.0033	0.0069	0.0233
1:20 (1:19)	ASCW1	0.0000	0.0000	0.0000	0.0542	0.0076	0.0354	0.0069	0.0024	0.0089
	ASCW2	0.0000	0.0023	0.0011	0.0357	0.0345	0.0501	0.0107	0.0019	0.0148

as $\hat{c}_+$. Then we compute the estimated errors as: $|c_+^* - \hat{c}_+|$ and present the results on Table 4.5. We can observe that the estimated errors of our algorithm and the optimal one is very close with the search length of 0.05, thus verifying the accurate estimation of our algorithm for the optimal cost.

4.6 Conclusion

Many real-world applications process data in an online-batch manner and suffer from the skewed distribution. In this chapter, we propose an adaptive sparse CW algorithm to deal with the feature selection problem on imbalanced online-batch data. Our algorithm simultaneously learns multiple base classifiers with their own costs. With the data comes sequentially in each online-batch, the aimed measure is updated incrementally for each classifier in an online-batch manner. Among all the classifiers, we choose the one with the best performance for prediction. We theoretically enhance the theory of the existing sparse CW feature selection algorithm and analyze the performance behavior regarding F-measure. Experimental results show the superior performance of ASCW and its ability for selecting the satisfactory cost vector.

Chapter 5

LSMI-Sinkhorn: Semi-supervised Squared-Loss Mutual Information Estimation with Optimal Transport

5.1 Introduction

Mutual information (MI) represents the statistical independence between two random variables [131], and it is widely used in various types of machine learning applications including feature selection [67, 66], dimensionality reduction [132], and causal inference [133]. More recently, deep neural network (DNN) models have started using MI as a regularizer for obtaining better representations from data such as infoVAE [134] and deep infoMax [135]. Another application is improving the generative adversarial networks (GANs) [136]. For instance, Mutual Information Neural Estimation (MINE) [4] was proposed to maximize or minimize the MI in deep networks and alleviate the mode-dropping issues in GANS. In all these examples, MI estimation is the core of all these applications.

In various MI estimation approaches, the probability density ratio function $r(\boldsymbol{x}, \boldsymbol{y})$is considered to be one of the most important components:

$$\mathrm{MI}(\boldsymbol{x}, \boldsymbol{y}) = \int\int p(\boldsymbol{x}, \boldsymbol{y}) \log \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}, \quad r(\boldsymbol{x}, \boldsymbol{y}) = \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}.$$

A straightforward method to estimate this ratio is the estimation of the probability densities (i.e., $p(\boldsymbol{x}, \boldsymbol{y})$, $p(\boldsymbol{x})$, and $p(\boldsymbol{y})$), followed by calculating their ratio. However, directly estimating the probability density is difficult, thereby making this two-step approach inefficient. To address the issue, Suzuki *et al.* [66] proposed to directly estimate the density ratio by avoiding the density estimation [67, 66]. Nonetheless, the abovementioned methods requires a large number of paired data when estimating the MI.

Under practical setting, we can only obtain a small number of paired samples. For example, it requires a massive amount of human labor to obtain one-to-one correspondences from one language to another. Thus, it prevents us to easily measure the MI across languages. Hence, a research question arises:

> *Can we perform mutual information estimation using unpaired samples and a small number of data pairs?*

To answer the above question, in this chapter, we propose a semi-supervised MI estimation algorithm, particularly designed for the squared-loss mutual information (SMI) (a.k.a., χ^2-divergence between $p(\boldsymbol{x}, \boldsymbol{y})$ and $p(\boldsymbol{x})p(\boldsymbol{y})$) [67]. We first formulate the SMI estimation as the optimal transport problem with density-ratio estimation. Then, we propose the least-squares mutual information with Sinkhorn (LSMI-Sinkhorn) algorithm to solve the problem. The algorithm has the computational complexity of $O(n_x n_y)$; hence, it is computationally efficient. Through experiments, we first demonstrate that the proposed method can estimate the SMI without a large number of paired samples. Finally, for image matching and photo album summarization, we show the effectiveness of our proposed method.

We summarize the contributions of this chapter as follows:

- We proposed a semi-supervised mutual information estimation approach that does not require a large number of paired samples.
- We formulated the MI estimation as a combination of density-ratio fitting and optimal transport.
- We proposed the LSMI-Sinkhorn algorithm, which can be efficiently computed and the loss is guaranteed to be monotonically decreasing.

5.2 Problem Formulation

In this section, we formulate the problem of squared-loss mutual information (SMI) estimation using a small number of paired samples and a large number of unpaired samples.

Let $\mathcal{X} \subset \mathbb{R}^{d_x}$ be the domain of random variable $\boldsymbol{x}$ and $\mathcal{Y} \subset \mathbb{R}^{d_y}$ be the domain of another random variable $\boldsymbol{y}$. Suppose we are given n independent and identically distributed (i.i.d.) *paired* samples:

$$\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n},$$

where we consider the number of paired samples n is small. In addition to the paired samples, we also have access to n_x and n_y i.i.d. samples from the marginal distributions:

$$\{\boldsymbol{x}_i\}_{i=n+1}^{n+n_x} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{x}) \text{ and } \{\boldsymbol{y}_j\}_{j=n+1}^{n+n_y} \overset{\text{i.i.d.}}{\sim} p(\boldsymbol{y}),$$

where the number of unpaired samples n_x and n_y is much larger than that of the paired samples n. For instance, $n = 10$ and $n_x = n_y = 1000$.

We also denote $\boldsymbol{x}'_i = \boldsymbol{x}_{i-n}, i \in \{n+1, n+2, \ldots, n+n_x\}$ and $\boldsymbol{y}'_j = \boldsymbol{y}_{j-n}, j \in \{n+1, n+2, \ldots, n+n_y\}$, respectively. Note that the input dimensions d_x, d_y and the number of samples n_x, n_y may be different.

This chapter aims to estimate the SMI (a.k.a., χ^2-divergence between $p(\boldsymbol{x}, \boldsymbol{y})$ and $p(\boldsymbol{x})p(\boldsymbol{y})$) [67] from $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$ by leveraging the use of the unpaired samples $\{\boldsymbol{x}_i\}_{i=n+1}^{n+n_x}$ and $\{\boldsymbol{y}_j\}_{j=n+1}^{n+n_y}$, respectively.

The SMI between random variables X and Y is defined as

$$\mathrm{SMI}(X, Y) = \frac{1}{2} \iint (r(\boldsymbol{x}, \boldsymbol{y}) - 1)^2 p(\boldsymbol{x}) p(\boldsymbol{y}) \mathrm{d}\boldsymbol{x} \mathrm{d}\boldsymbol{y}, \tag{5.1}$$

where $r(\boldsymbol{x}, \boldsymbol{y}) = \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}$ is the density-ratio function. SMI takes 0 if and only if X and Y are independent (i.e., $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x})p(\boldsymbol{y})$), and takes a positive value if they are not independent.

If we know the estimation of the density-ratio function, we can approximate the SMI as

$$\widehat{\mathrm{SMI}}(X,Y)=\frac{1}{2(n+n_x)(n+n_y)}\sum_{i=1}^{n+n_x}\sum_{j=1}^{n+n_y}(r_{\boldsymbol{\alpha}}(\boldsymbol{x}_i,\boldsymbol{y}_j)-1)^2\,,$$

where $r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})$ is an estimation of the true density ratio function (i.e., $r(\boldsymbol{x},\boldsymbol{y})$) parameterized by $\boldsymbol{\alpha}$. More details will be discussed in §5.3.1.

However, since we consider the setting that we lack enough paired samples to estimate the density ratio, which may result in high variance and bias when computing the SMI. The key idea is to align the unpaired samples when observing the limited number of paired samples, and we use these aligned samples to improve the SMI estimation accuracy.

5.3 Proposed Method

In this section, we propose the SMI estimation algorithm with limited number of paired samples and large number of unpaired samples.

5.3.1 Least-Squares Mutual Information with Sinkhorn (LSMI-Sinkhorn)

We employ the following density-ratio model, which we first sample two sets basis vectors $\{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{b}$ and $\{\widetilde{\boldsymbol{y}}_i\}_{i=1}^{b}$ from $\{\boldsymbol{x}_i\}_{i=1}^{n+n_x}$ and $\{\boldsymbol{y}_j\}_{j=1}^{n+n_y}$, respectively:

$$r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})=\sum_{\ell=1}^{b}\alpha_\ell K(\widetilde{\boldsymbol{x}}_\ell,\boldsymbol{x})L(\widetilde{\boldsymbol{y}}_\ell,\boldsymbol{y})=\boldsymbol{\alpha}^\top\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}), \tag{5.2}$$

where $\boldsymbol{\alpha}\in\mathbb{R}^b$, $K(\boldsymbol{x},\boldsymbol{x}')$ and $L(\boldsymbol{y},\boldsymbol{y}')$ are the kernel functions. $\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y})=\boldsymbol{k}(\boldsymbol{x})\circ\boldsymbol{l}(\boldsymbol{y})$ with $\boldsymbol{k}(\boldsymbol{x})=(K(\widetilde{\boldsymbol{x}}_1,\boldsymbol{x}),\ldots,K(\widetilde{\boldsymbol{x}}_b,\boldsymbol{x}))^\top\in\mathbb{R}^b$ and $\boldsymbol{l}(\boldsymbol{y})=(L(\widetilde{\boldsymbol{y}}_1,\boldsymbol{y}),\ldots,L(\widetilde{\boldsymbol{y}}_b,\boldsymbol{y}))^\top\in\mathbb{R}^b$.

In this chapter, we optimize $\boldsymbol{\alpha}$ by minimizing the difference between the true density-ratio function and its ratio model:

$$\frac{1}{2}\iint\left(\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}-r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})\right)^2p(\boldsymbol{x})p(\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}$$

$$
\begin{aligned}
&= \text{Const.} + \frac{1}{2}\iint r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})^2 p(\boldsymbol{x})p(\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}. \\
&- \iint r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}. \qquad (5.3)
\end{aligned}
$$

For the second term of Eq. (5.3), we can approximate it by using a large number of unpaired samples. However, to approximate the third term, paired samples from the joint distribution are required. Since we only have a limited number of paired samples in our setting, the approximation of the third term may be poor.

To deal with this issue, we propose the utilization of unpaired samples for the approximation of the expectation of the third term. Since we have no access to the true pair information for these unpaired samples, we approximate the pair information of them. Specifically, we first introduce a matrix $\boldsymbol{\Pi}$ with $\pi_{ij} \geq 0$ ($\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\pi_{i,j} = 1$) that can be regarded as a parameterized variant of the joint density function $p(\boldsymbol{x},\boldsymbol{y})$, and we represent the third term of Eq. (5.3) as

$$
\begin{aligned}
&\iint r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y} \\
&\approx \frac{\beta}{n}\sum_{i=1}^{n} r_{\boldsymbol{\alpha}}(\boldsymbol{x}_i,\boldsymbol{y}_i) + (1-\beta)\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\pi_{ij} r_{\boldsymbol{\alpha}}(\boldsymbol{x}'_i,\boldsymbol{y}'_j),
\end{aligned}
$$

where $0 \leq \beta \leq 1$ is a tuning parameter between the terms of paired and unpaired samples. Note that if we set $\pi_{ij} = \delta(\boldsymbol{x}'_i,\boldsymbol{y}'_j)/n'$ where $\delta(\boldsymbol{x}'_i,\boldsymbol{y}'_j)$ is one if $\boldsymbol{x}'_i$ and $\boldsymbol{y}'_j$ are paired and 0 otherwise, and n' is the total number of pairs, then we can recover the original empirical estimation (with no approximation for pair information of unpaired samples).

Combining the estimation for the density-ratio model (Eq. (5.2)) and the approximated pairing matrix ($\boldsymbol{\Pi}$), the loss function in Eq. (5.3) can be reformulated as

$$
J(\boldsymbol{\Pi},\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \boldsymbol{h}_{\boldsymbol{\Pi},\beta},
$$

where

$$\boldsymbol{H} = \frac{1}{(n+n_x)(n+n_y)} \sum_{i=1}^{n+n_x} \sum_{j=1}^{n+n_y} \boldsymbol{\varphi}(\boldsymbol{x}_i, \boldsymbol{y}_j)\boldsymbol{\varphi}(\boldsymbol{x}_i, \boldsymbol{y}_j)^\top,$$

$$\boldsymbol{h}_{\boldsymbol{\Pi},\beta} = \frac{\beta}{n} \sum_{i=1}^{n} \boldsymbol{\varphi}(\boldsymbol{x}_i, \boldsymbol{y}_i) + (1-\beta) \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \pi_{ij} \boldsymbol{\varphi}(\boldsymbol{x}'_i, \boldsymbol{y}'_j).$$

Since we want to estimate the density-ratio function by minimizing Eq. (5.3), the optimization problem is then given as

$$\begin{aligned} \min_{\boldsymbol{\Pi},\boldsymbol{\alpha}} \quad & J(\boldsymbol{\Pi}, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \boldsymbol{h}_{\boldsymbol{\Pi},\beta} + \epsilon H(\boldsymbol{\Pi}) + \frac{\lambda}{2}\|\boldsymbol{\alpha}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\Pi}\mathbf{1}_{n_y} = n_x^{-1}\mathbf{1}_{n_x} \text{ and } \boldsymbol{\Pi}^\top \mathbf{1}_{n_x} = n_y^{-1}\mathbf{1}_{n_y}. \end{aligned} \tag{5.4}$$

where we add several regularization terms. $H(\boldsymbol{\Pi}) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \pi_{ij}(\log \pi_{ij} - 1)$ is the negative entropic regularization to ensure $\boldsymbol{\Pi}$ non-negative with $\epsilon > 0$ being its regularization parameter. $\|\boldsymbol{\alpha}\|_2^2$ is the regularization on $\boldsymbol{\alpha}$ with $\lambda \geq 0$ being its regularization parameter.

5.3.2 Optimization

The objective function $J(\boldsymbol{\Pi}, \boldsymbol{\alpha})$ is not jointly convex. However, if we fix one variable, it becomes a convex function for the other. Thus, we employ the alternating optimization approach (see Algorithm 3) on $\boldsymbol{\Pi}$ and $\boldsymbol{\alpha}$, respectively.

Optimizing $\boldsymbol{\Pi}$ using the Sinkhorn algorithm: When fixing $\boldsymbol{\alpha}$, the term in our objective relating to $\boldsymbol{\Pi}$ is

$$\sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \pi_{ij} \boldsymbol{\alpha}^\top \boldsymbol{\varphi}(\boldsymbol{x}'_i, \boldsymbol{y}'_j) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \pi_{ij} [\boldsymbol{C}_{\boldsymbol{\alpha}}]_{ij},$$

where $\boldsymbol{C}_{\boldsymbol{\alpha}} = \boldsymbol{K}^\top \text{diag}(\boldsymbol{\alpha})\boldsymbol{L} \in \mathbb{R}^{n_x \times n_y}$, $\boldsymbol{K} = (\boldsymbol{k}(\boldsymbol{x}'_1), \boldsymbol{k}(\boldsymbol{x}'_2), \ldots, \boldsymbol{k}(\boldsymbol{x}'_{n_x})) \in \mathbb{R}^{b \times n_x}$, and $\boldsymbol{L} = (\boldsymbol{l}(\boldsymbol{y}'_1), \boldsymbol{l}(\boldsymbol{y}'_2), \ldots, \boldsymbol{l}(\boldsymbol{y}'_{n_y})) \in \mathbb{R}^{b \times n_y}$. This formulation can be considered as an optimal transport problem if we maximize it with respect to $\boldsymbol{\Pi}$ [72]. It is worth noting that the rank of $\boldsymbol{C}_\alpha$ is at most $b \ll \min(n_x, n_y)$ with b being a constant (e.g.,

Algorithm 3 LSMI-Sinkhorn algorithm.

```
Initialize $\boldsymbol{\Pi}^{(0)}$ and $\boldsymbol{\Pi}^{(1)}$ such that $\|\boldsymbol{\Pi}^{(1)} - \boldsymbol{\Pi}^{(0)}\|_F > \eta$ ($\eta$ is the stopping parameter), and $\boldsymbol{\alpha}^{(0)}$, the regularization parameters $\epsilon$ and $\lambda$, the number of maximum iterations $T$, and the iteration index $t = 1$.
while $t \leq T$ and $\|\boldsymbol{\Pi}^{(t)} - \boldsymbol{\Pi}^{(t-1)}\|_F > \eta$ do
    $\boldsymbol{\alpha}^{(t+1)} = \mathrm{argmin}_{\boldsymbol{\alpha}} J(\boldsymbol{\Pi}^{(t)}, \boldsymbol{\alpha})$.
    $\boldsymbol{\Pi}^{(t+1)} = \mathrm{argmin}_{\boldsymbol{\Pi}} J(\boldsymbol{\Pi}, \boldsymbol{\alpha}^{(t+1)})$.
    $t = t + 1$.
end while
return $\boldsymbol{\Pi}^{(t-1)}$ and $\boldsymbol{\alpha}^{(t-1)}$.
```

$b = 100$), and the computational complexity of the cost matrix $\boldsymbol{C}_{\boldsymbol{\alpha}}$ is $O(n_x n_y)$. The optimization problem becomes

$$\begin{aligned} \min_{\boldsymbol{\Pi}} \quad & -\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} \pi_{ij}(1-\beta)[\boldsymbol{C}_{\boldsymbol{\alpha}}]_{ij} + \epsilon H(\boldsymbol{\Pi}) \\ \text{s.t.} \quad & \boldsymbol{\Pi}\boldsymbol{1}_{n_y} = n_x^{-1}\boldsymbol{1}_{n_x} \text{ and } \boldsymbol{\Pi}^\top\boldsymbol{1}_{n_x} = n_y^{-1}\boldsymbol{1}_{n_y}, \end{aligned}$$

which can be efficiently solved using the Sinkhorn algorithm [72, 137]. In this chapter, we use the log-stabilized Sinkhorn [138]. Note that this optimization problem is convex with fixed $\boldsymbol{\alpha}$.

Optimizing $\boldsymbol{\alpha}$: Next, we fix $\boldsymbol{\Pi}$ and update $\boldsymbol{\alpha}$. The optimization problem becomes

$$\min_{\boldsymbol{\alpha}} \ \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \boldsymbol{h}_{\boldsymbol{\Pi},\beta} + \frac{\lambda}{2}\|\boldsymbol{\alpha}\|_2^2, \tag{5.5}$$

which is a quadratic programming and convex. An analytical solution is given as

$$\widehat{\boldsymbol{\alpha}} = (\boldsymbol{H} + \lambda \boldsymbol{I}_b)^{-1}\boldsymbol{h}_{\boldsymbol{\Pi},\beta}, \tag{5.6}$$

where $\boldsymbol{I}_b \in \mathbb{R}^{b\times b}$ is an identity matrix. Note that the $\boldsymbol{H}$ matrix does not depend on either $\boldsymbol{\Pi}$ or $\boldsymbol{\alpha}$, and it is a positive definite matrix.

Convergence Analysis: To optimize $J(\mathbf{\Pi}, \boldsymbol{\alpha})$, we simply need to alternatively solve the two convex optimization problems. Thus, the following property holds true.

Proposition 2. Algorithm 3 will monotonically decrease the objective function $J(\mathbf{\Pi}, \boldsymbol{\alpha})$ in each iteration.

Proof. We show that $J(\mathbf{\Pi}^{(t+1)}, \boldsymbol{\alpha}^{(t+1)}) \leq J(\mathbf{\Pi}^{(t)}, \boldsymbol{\alpha}^{(t)})$. First, because $\boldsymbol{\alpha}^{(t+1)} = \operatorname{argmin}_{\boldsymbol{\alpha}} J(\mathbf{\Pi}^{(t)}, \boldsymbol{\alpha})$ and $\boldsymbol{\alpha}^{(t+1)}$ is the globally optimum solution, we have

$$J(\mathbf{\Pi}^{(t)}, \boldsymbol{\alpha}^{(t+1)}) \leq J(\mathbf{\Pi}^{(t)}, \boldsymbol{\alpha}^{(t)}).$$

Moreover, because $\mathbf{\Pi}^{(t+1)} = \operatorname{argmin}_{\mathbf{\Pi}} J(\mathbf{\Pi}, \boldsymbol{\alpha}^{(t+1)})$ and $\mathbf{\Pi}^{(t+1)}$ is the globally optimum solution, we have

$$J(\mathbf{\Pi}^{(t+1)}, \boldsymbol{\alpha}^{(t+1)}) \leq J(\mathbf{\Pi}^{(t)}, \boldsymbol{\alpha}^{(t+1)}).$$

Therefore,

$$J(\mathbf{\Pi}^{(t+1)}, \boldsymbol{\alpha}^{(t+1)}) \leq J(\mathbf{\Pi}^{(t)}, \boldsymbol{\alpha}^{(t)}).$$

□

Model Selection: We name Algorithm 3 as LSMI-Sinkhorn algorithm since it utilizes Sinkhorn algorithm for LSMI estimation. It includes several tuning parameters (i.e., λ and β) and determining the model parameters is critical to obtain a good estimate of SMI. Accordingly, we use the cross validation with the hold-out set to select the model parameters.

First, the paired samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$ are divided into two subsets $\mathcal{D}_{\mathrm{tr}}$ and $\mathcal{D}_{\mathrm{te}}$. Then, we train the density-ratio $r_{\boldsymbol{\alpha}}(\boldsymbol{x}, \boldsymbol{y})$ using $\mathcal{D}_{\mathrm{tr}}$ and the unpaired samples: $\{\boldsymbol{x}_i\}_{i=n+1}^{n+n_x}$ and $\{\boldsymbol{y}_j\}_{j=n+1}^{n+n_y}$. The hold-out error can be calculated by approximating Eq. (5.3) using the hold-out samples $\mathcal{D}_{\mathrm{te}}$ as

$$\widehat{J}_{\mathrm{te}} = \frac{1}{2|\mathcal{D}_{\mathrm{te}}|^2} \sum_{\boldsymbol{x}, \boldsymbol{y} \in \mathcal{D}_{\mathrm{te}}} r_{\widehat{\boldsymbol{\alpha}}}(\boldsymbol{x}, \boldsymbol{y})^2 - \frac{1}{|\mathcal{D}_{\mathrm{te}}|} \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}_{\mathrm{te}}} r_{\widehat{\boldsymbol{\alpha}}}(\boldsymbol{x}, \boldsymbol{y}),$$

where $|\mathcal{D}|$ denotes the number of samples in the set $\mathcal{D}$, $\sum_{\boldsymbol{x},\boldsymbol{y}\in\mathcal{D}_{\mathrm{te}}}$ denotes the summation over all combinations of $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathcal{D}_{\mathrm{te}}$, and $\sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}_{\mathrm{te}}}$ denotes the summation over all pairs for $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathcal{D}_{\mathrm{te}}$. We select the parameters that result in the smallest $\widehat{J}_{\mathrm{te}}$.

5.3.3 Discussion

Relation to Least-Squares Object Matching (LSOM): In this section, we show that the LSOM algorithm [139, 140] can be considered as a special case of the proposed framework.

If $\boldsymbol{\Pi}$ is a permutation matrix and $n' = n_x = n_y$,

$$\boldsymbol{\Pi} = \{0,1\}^{n'\times n'},\ \boldsymbol{\Pi}\boldsymbol{1}_{n'} = \boldsymbol{1}_{n'},\ \text{and}\ \boldsymbol{\Pi}^\top\boldsymbol{1}_{n'} = \boldsymbol{1}_{n'},$$

where $\boldsymbol{\Pi}^\top\boldsymbol{\Pi} = \boldsymbol{\Pi}\boldsymbol{\Pi}^\top = \boldsymbol{I}_{n'}$.

Then, the estimation of SMI using the permutation matrix can be written as

$$\begin{aligned}&\widehat{\mathrm{SMI}}(X,Y)\\&=\frac{\beta}{2n}\sum_{i=1}^{n} r_{\boldsymbol{\alpha}}(\boldsymbol{x}_i,\boldsymbol{y}_i)+\frac{1}{2n'}\sum_{i=1}^{n'}(1-\beta)r_{\boldsymbol{\alpha}}(\boldsymbol{x}'_i,\boldsymbol{y}'_{\pi(i)})-\frac{1}{2},\end{aligned}$$

where $\pi(i)$ is the permutation function. The optimization problem is written as

$$\begin{aligned}&\min_{\boldsymbol{\Pi},\boldsymbol{\alpha}}\ \frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{H}\boldsymbol{\alpha}-\boldsymbol{\alpha}^\top\boldsymbol{h}_{\boldsymbol{\Pi},\beta}+\frac{\lambda}{2}\|\boldsymbol{\alpha}\|_2^2\\&\ \text{s.t.}\ \ \boldsymbol{\Pi}\boldsymbol{1}_{n'}=\boldsymbol{1}_{n'},\ \boldsymbol{\Pi}^\top\boldsymbol{1}_{n'}=\boldsymbol{1}_{n'},\ \boldsymbol{\Pi}\in\{0,\ 1\}^{n'\times n'}.\end{aligned}$$

To solve this problem, we can use the Hungarian algorithm [141] instead of the Sinkhorn algorithm [72] for optimizing $\boldsymbol{\Pi}$. It is noteworthy that in the original LSOM algorithm, the permutation matrix is introduced to permute the Gram matrix (i.e., $\boldsymbol{\Pi}\boldsymbol{L}\boldsymbol{\Pi}^\top$) and $\boldsymbol{\Pi}$ is also included within the $\boldsymbol{H}$ computation. However, in our formulation, the permutation matrix depends only on $\boldsymbol{h}_{\boldsymbol{\Pi},\beta}$. This difference enables us to show a monotonic decrease in the loss function of the proposed algorithm.

Since LSOM aims to find the alignment, it is more suited to find the exact matching among samples. In contrast, the proposed formulation is more suited when there are no exact matches. Moreover, the LSOM formulation assumes the same number of samples (i.e., $n_x = n_y$), while our approach does not have this constraint. For computational complexity, the Hungarian algorithm requires $O({n'}^3)$ while the Sinkhorn requires $O({n'}^2)$.

Computational Complexity: The computational complexity of estimating $\mathbf{\Pi}$ is based on the computation of the cost matrix $\boldsymbol{C_\alpha}$ and the Sinkhorn iterations. The computational complexity of $\boldsymbol{C_\alpha}$ is $O(n_x n_y)$ and that of Sinkhorn algorithm is $O(n_x n_y)$. Therefore, the computational complexity of the Sinkhorn iteration is $O(n_x n_y)$. For the $\boldsymbol{\alpha}$ computation, the complexity to compute $\boldsymbol{H}$ is $O((n + n_x)^2 + (n + n_y)^2)$ and that for $\boldsymbol{h}_{\mathbf{\Pi},\beta}$ is $O(n_x n_y)$. Although estimating $\boldsymbol{\alpha}$ has complexity $O(b^3)$, the small-valued constant b makes it negligible. To conclude, the total computational complexity of the initialization needs $O((n + n_x)^2 + (n + n_y)^2)$ and the iterations requires $O(n_x n_y)$. In particular, for small n and $n_x = n_y$, the computational complexity is $O(n_x^2)$.

In contrast, the complexity of computing the objective function of Gromov-Wasserstein is $O(n_x^4)$ for general cases and $O(n_x^3)$ for some specific losses (e.g. L_2 loss, or Kullback-Leibler loss) [70]. Moreover, Gromov-Wasserstein is generally NP-hard for arbitrary inputs [70, 71].

5.4 Experiments

In this section, we evaluate the proposed algorithm using the synthetic data and benchmark datasets.

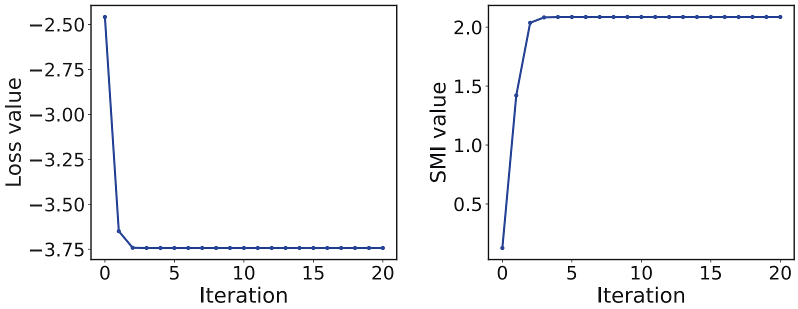


Figure 5.1 : Convergence curves of the loss and SMI values.

5.4.1 Setup

For all methods, we use the Gaussian kernels: $K(\boldsymbol{x},\boldsymbol{x}')=\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2}{2\sigma_x^2}\right)$, $L(\boldsymbol{y},\boldsymbol{y}')=\exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{y}'\|_2^2}{2\sigma_y^2}\right)$, where σ_x and σ_y denote the widths of the kernel that are set using the median heuristic [142]. $\sigma_x = 2^{-1/2}\text{median}(\{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2\}_{i,j=1}^{n_x}), \sigma_y = 2^{-1/2}\text{median}(\{\|\boldsymbol{y}_i-\boldsymbol{y}_j\|_2\}_{i,j=1}^{n_y})$. We set the number of basis $b = 200$, $\epsilon = 0.3$, the maximum number of iterations $T = 20$, and the stopping parameter $\eta = 10^{-9}$. The parameters β and λ are chosen by cross-validation.

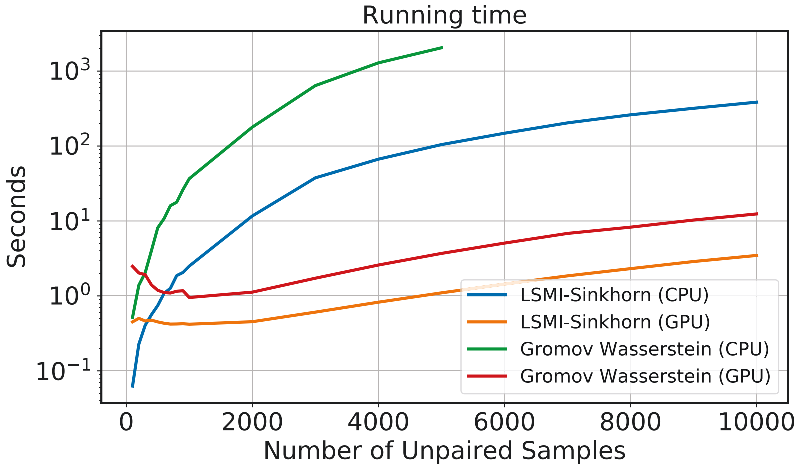


Figure 5.2 : Runtime of LSMI-Sinkhorn and Gromov-Wasserstein. A base-10 log scale is used for the Y axis.

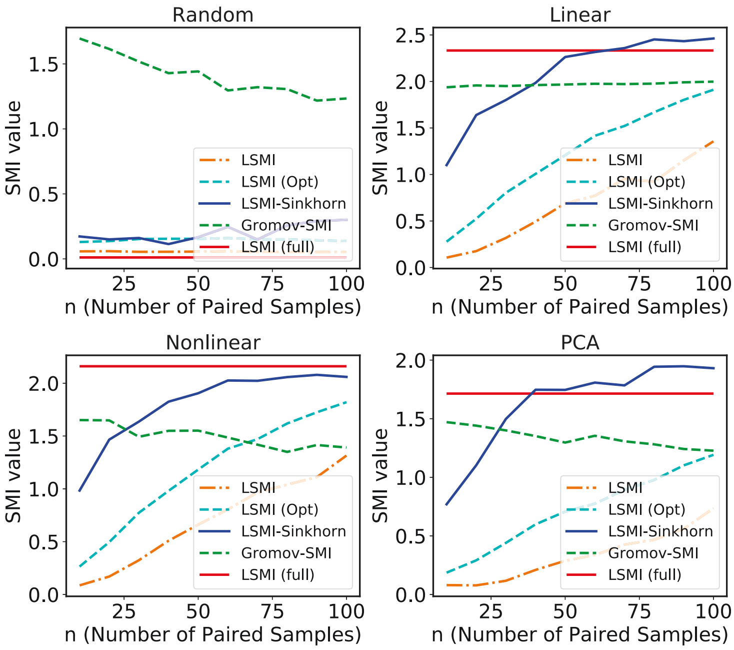


Figure 5.3 : SMI estimation on synthetic data ($n_x = n_y = 500$).

5.4.2 Convergence and Runtime

We first demonstrate the convergence of the loss function and the estimated SMI value. Here, we generate synthetic data from $\boldsymbol{y} = 0.5\boldsymbol{x} + \mathcal{N}(0, 0.01)$ and randomly choose $n = 50$ paired samples and $n_x = n_y = 500$ unpaired samples. The convergence curve is shown in Figure 5.1. The values of loss and SMI converge quickly (<5 iterations). This is consistent with Proposition 2.

Then, we perform a comparison between the runtimes of the proposed LSMI-Sinkhorn and Gromov-Wasserstein for CPU and GPU implementation. The data are sampled from two 2D random measures, where $n_x = n_y \in \{100, 200, \ldots, 9000, 10000\}$ is the number of unpaired data and $n = 100$ is the number of paired data (only for LSMI-Sinkhorn). For Gromov-Wasserstein, we use the CPU implementation from Python Optimal Transport toolbox [143] and the Pytorch GPU implementation from [96]. We use the squared loss function and set the entropic regularization ϵ to 0.005 according to the original code. For LSMI-Sinkhorn, we implement the CPU

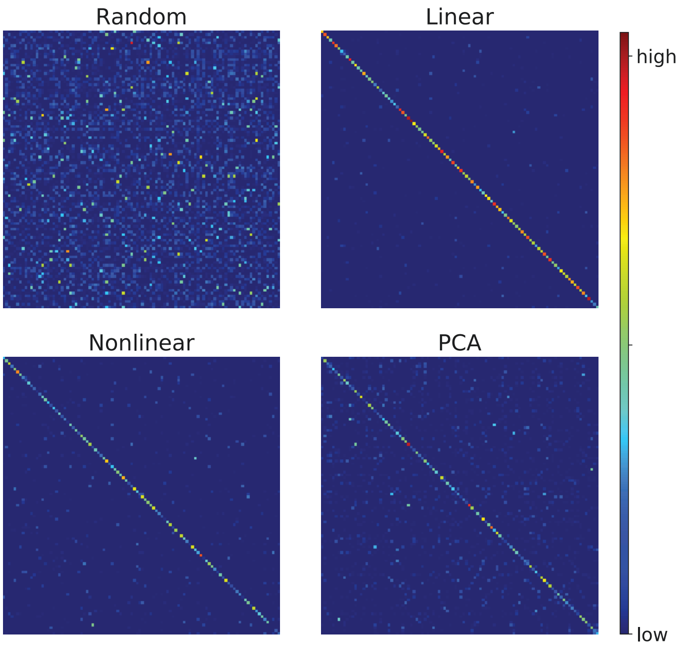


Figure 5.4 : Visualization of the matrix $\mathbf{\Pi}$.

and GPU versions using numpy and Pytorch, respectively. For fair comparison, we use the log-stabilized Sinkhorn algorithm and the same early stopping criteria and the same maximum iterations as in Gromov-Wasserstein. As shown in Figure 5.2, in comparison to the Gromov-Wasserstein, LSMI-Sinkhorn is more than one order of magnitude faster for the CPU version and several times faster for the GPU version. This is consistent with our computational complexity analysis. Moreover, the GPU version of our algorithm costs only 3.47s to compute $10,000$ unpaired samples, indicating that it is suitable for large-scale applications.

5.4.3 SMI Estimation

For SMI estimation, we set up four baselines:

- **LSMI (full)**: $10,000$ paired samples are used for cross-validation and SMI estimation. It is considered as the ground truth value.
- **LSMI**: Only n (usually small) paired samples are used for cross-validation and

SMI estimation.

- **LSMI (opt)**: n paired samples are used for SMI estimation. However, we use the optimal parameters from LSMI (full) here. This can be seen as the upper bound of SMI estimation with limited number of paired data because the optimal parameters are usually unavailable.

- **Gromov-SMI**: The Gromov-Wasserstein distance is applied on unpaired samples to find potential matching ($\hat{n} = \min(n_x, n_y)$). Then, the $\hat{n}$ matched pairs and existing n paired samples are combined to perform cross-validation and SMI estimation.

Synthetic Data: In this experiment, we manually generate four types of paired samples: random normal, $\boldsymbol{y} = 0.5\boldsymbol{x} + \mathcal{N}(0, 0.01)$ (Linear), $\boldsymbol{y} = \sin(\boldsymbol{x})$ (Nonlinear), and $\boldsymbol{y} = \text{PCA}(\boldsymbol{x})$. We change the number of paired samples $n \in \{10, 20, \dots, 100\}$ while fixing $n_x = 500$ and $n_y = 500$ for Gromov-SMI and the proposed LSMI-Sinkhorn, respectively. The model parameters λ and β are selected by cross-validation using the paired examples with $\lambda \in \{0.1, 0.01, 0.001, 0.0001\}$ and $\beta \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$. The results are shown in Figure 5.3. In the random case, the data are nearly independent and our algorithm achieves a small SMI value. In other cases, LSMI-Sinkhorn yields a better estimation of the SMI value and it lies near the ground truth when n increases. In contrast, Gromov-SMI has a small estimation value, which may be due to the incorrect potential matching. We further show the heatmaps of the matrix $\boldsymbol{\Pi}$ in Figure 5.4. For the random case, $\boldsymbol{\Pi}$ distributes uniformly as expected. For all other cases, $\boldsymbol{\Pi}$ concentrate on the diagonal, indicating good estimation for the unpaired samples.

To show the flexibility of the proposed LSMI-Sinkhorn algorithm, we set $n_x = 1000, n_y = 500$ and fix all other settings. The results are shown in Figure 5.5. Similarly, LSMI-Sinkhorn achieves the best performance among all methods. We

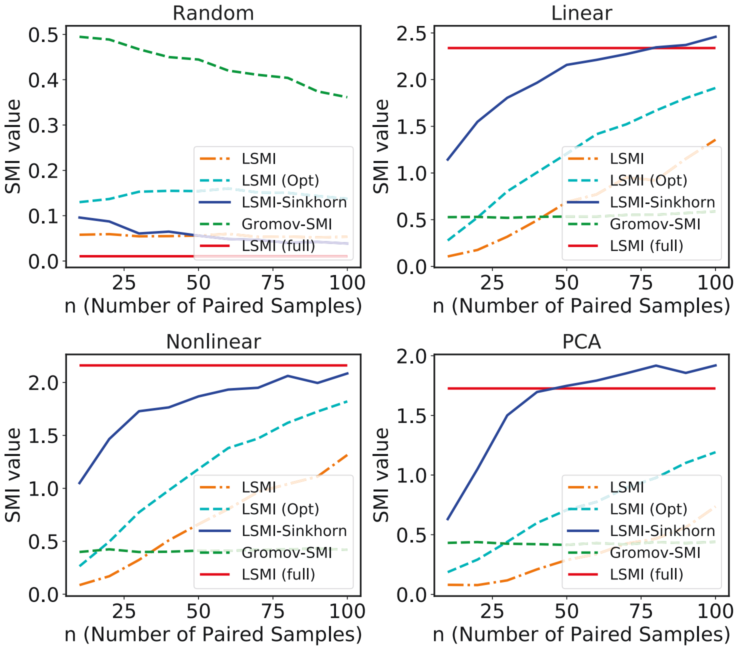


Figure 5.5 : SMI estimation on synthetic data ($n_x = 1000, n_y = 500$).

also notice that Gromov-SMI achieves even worse estimation than $n_x = n_y$ case, which means it is not as stable as our algorithm to handle sophisticated situations ($n_x \neq n_y$).

UCI Datasets: We selected four benchmark datasets from the UCI machine learning repository. For each dataset, we split the features into two sets as paired samples. To ensure high dependence between these two subsets of features, we utilized the same splitting strategy as [74] according to the correlation matrix. The experimental setting is the same as the synthetic data experiment. We show the SMI estimation results in Figure 5.6. Similarly, LSMI-Sinkhorn obtains better estimation values in all four datasets. Gromov-SMI tends to overestimate the value by a large margin, while other baselines underestimate the value.

5.4.4 Deep Image Matching

Next, we consider an image matching task with deep convolution features. We use two commonly-used image classification benchmarks: CIFAR10 [144] and STL10

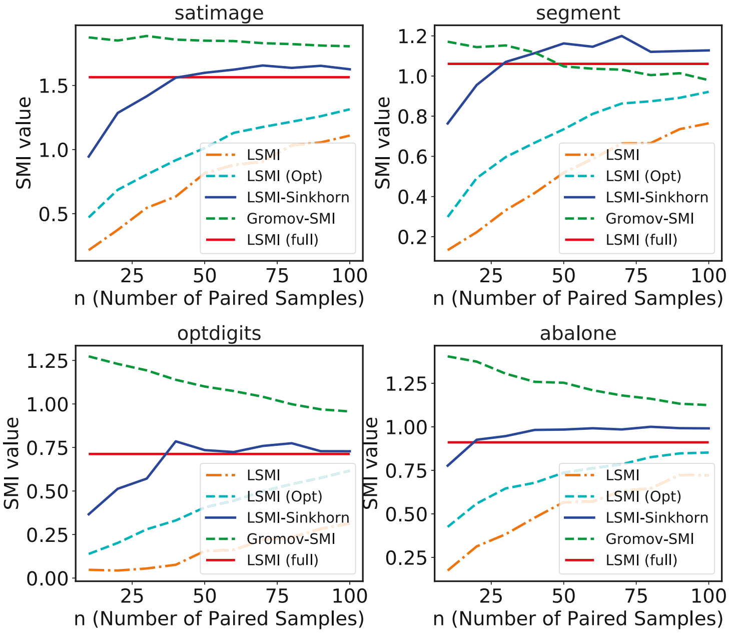


Figure 5.6 : SMI estimation on UCI datasets.

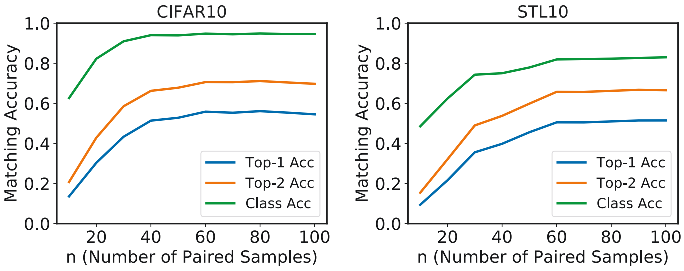


Figure 5.7 : Deep image matching.

[145].We extracted 64-dim features from the last layer (after pooling) of ResNet20 [7] pretrained on the training set of CIFAR10. The features are divided into two 32-dim parts denoted by $\{\boldsymbol{x}_i\}_{i=1}^N$ and $\{\boldsymbol{y}_i\}_{i=1}^N$. We shuffle the samples of $\boldsymbol{y}$ and attempt to match $\boldsymbol{x}$ and $\boldsymbol{y}$ with limited pair samples ($n \in \{10, 20, \ldots, 100\}$) and unpaired samples ($n_x = n_y = 500$). Other settings are the same as the above experiments.

To evaluate the matching performance, we used top-1 accuracy, top-2 accuracy (correct matching is achieved in the top-2 highest scores), and class accuracy (matched samples are in the same class). As shown in Figure 5.7, LSMI-Sinkhorn obtains high accuracy with only a few tens of supervised pairs. Additionally, the high class matching performance implies that our algorithm can be applied to further applications such as semi-supervised image classification.

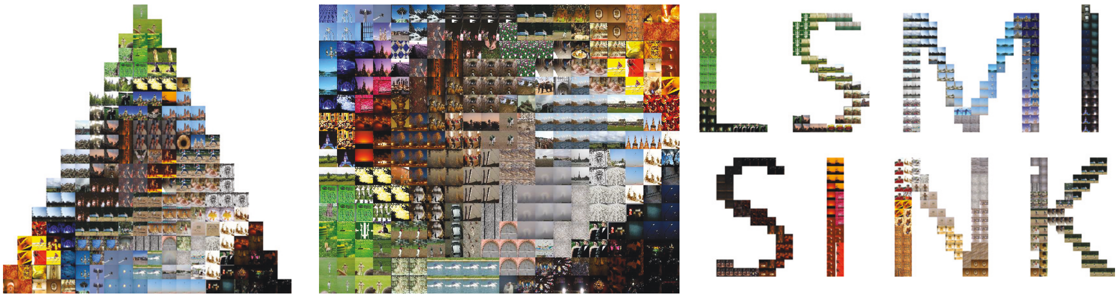
5.4.5 Photo Album Summarization

Finally, we apply the proposed LSMI-Sinkhorn to the photo album summarization problem, where images are matched to a predefined structure according to the Cartesian coordinate system.

Color Feature. We first used 320 images collected from Flickr [74] and extracted the original RGB pixels as color feature. Figure 5.8a and 5.8b depict the semi-supervised summarization to the triangle and 16×20 grids with the corners of the grids fixed to *green*, *orange*, *black* (triangle), and *blue* (rectangle) images. Similarly, we show the summarization results on an "LSMI SINK" grid with the center of each character fixed. It can be seen that these layouts show good color topology according to the fixed color images.

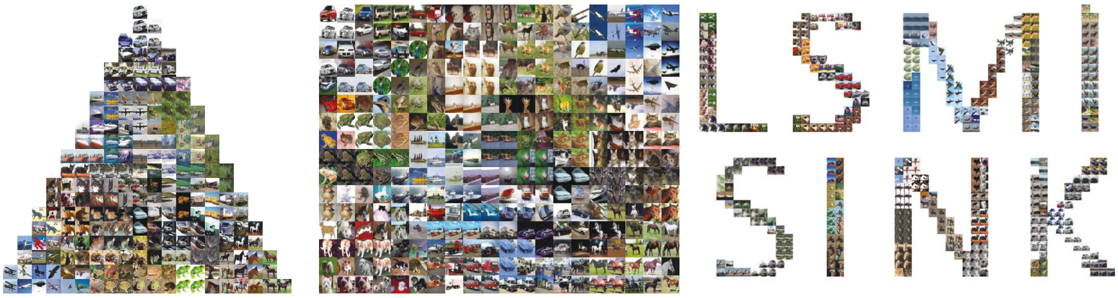
Semantic Feature. We then used CIFAR10 with the ResNet20 feature to illustrate the semantic album summarization. Figure 5.9 shows the layout of 1000 images into the same triangle, 16×20, and "LSMI SINK" grids. For Figure 5.9a and 5.9b, we fixed corners of the grid to *automobile*, *airplane*, *horse* (triangle) and *dog* (rectangle) images. For Figure 5.9c, we fixed the corresponding character centers. It can be seen that similar objects are aligned together by their semantic meanings rather than colors of the fixed images.

In comparison to previous summarization algorithms, LSMI-Sinkhorn has two advantages. First, the semi-supervised property enables interactive album summarization, while kernelized sorting [74, 75] and object matching [139] cannot. Second, we obtained a solution for general rectangular matching ($n_x \neq n_y$), e.g., 320 images to a triangle grid, 1000 images to a 16×20 grid, while most previous methods [74, 139] relied on the Hungarian algorithm [141] to obtain square matching ($n_x = n_y$).



(a) Triangle (b) 16 × 20 (c) “LSMI SINK"

Figure 5.8 : Photo album summarization on Flickr dataset. For (a) and (b), we fixed the corners with *green*, *orange*, *black* (triangle), and *blue* (rectangle) images. For (c), we fixed the center of each character with different images.



(a) Triangle (b) 16 × 20 (c) “LSMI SINK"

Figure 5.9 : Photo album summarization on CIFAR10 dataset. For (a) and (b), we fixed the corners with *automobile*, *airplane*, *horse* (triangle), and *dog* (rectangle) images. For (c), we fixed the center of each character with different images.

5.5 Conclusion

In this chapter, we proposed the LSMI-Sinkhorn algorithm to estimate the SMI from a limited number of paired samples. To the best of our knowledge, this is the first semi-supervised SMI estimation algorithm. Experiments on synthetic and real data showed that the proposed algorithm can successfully estimate SMI with a small number of paired samples. Moreover, we demonstrated that the proposed algorithm can be used for image matching and photo album summarization.

Chapter 6

Semantic Correspondence as an Optimal Transport Problem

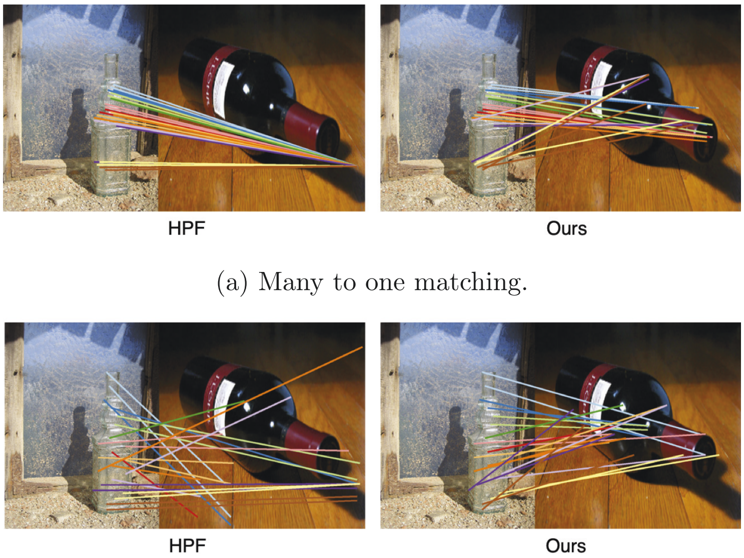
6.1 Introduction

Establishing dense correspondences across semantically similar images is one of the fundamental tasks in computer vision that has potential applications such as semantic segmentation [146, 147], image registration [89], and image editing [148, 149]. This is a challenging task due to the large intra-class variation, viewpoint changes and background clutter.

Recent methods employ powerful image features from convolutional neural networks. Semantic flow approaches attempt to establish a flow field between images based on single [83, 3, 2] or multiple layers [1] feature maps. Semantic alignment methods cast semantic correspondence as a geometric alignment problem to regress the global transformation parameters using self-supervised [87, 88], weakly-supervised [150, 3] or keypoints [89] supervision. However, *many to one matching* problem and *background matching* problem hinder the development of semantic correspondence.

First, *many to one matching* occurs when many pixels in a source image are assigned to one target pixel. We solve this problem by global feature matching, which maximizes the total matching correlations between images. Most existing approaches [83, 84, 1, 3] for semantic correspondence rely on the correlation map which is computed by individual feature matching*. The individual matching scheme does

*Each source feature finds its nearest neighbor from all target features independently.



(a) Many to one matching.

(b) Background matching.

Figure 6.1 : We solve two problems caused by current approaches, such as HPF [1]. (a) Many pixels in a source image are assigned to one target pixel. (b) Some object pixels are assigned to the background pixels. Note that the two different results are only part of the whole results which reflect the many to one matching and background matching.

not care about the mutual relation between features within the same image. Therefore, it is sensitive to large intra-class variations and repetitive patterns (*i.e.*, similar patterns for different parts of an object.). For example, in Figure 6.1a (Left), due to repetitive pattern in left bottle, the individual matching assigns many source pixels to one target pixel. Although, semantic alignment methods [150, 89, 88] try to suppress many to one matching by estimating the global transformation parameters, they are easily distracted by occlusion and non-rigid deformations. In our method, maximizing the total matching correlations leads to a global optimal matching matrix, which is insensitive to repetitive patterns (e.g., Figure 6.1a (Right)). For the matching matrix, each row represents matching scores from a source pixel to all target pixels and each column represents scores from all source pixels to a target pixel. We enforce each row sum and column sum to be a fixed value according to the prior distributions of pixels. This avoids large values in a whole row or column, thus reducing the many to one matching.

Second, *background matching* happens when some object pixels are assigned to background pixels due to the intra-class appearance variation and background clutter, as shown in Figure 6.1b (Left). Recent methods deal with this by soft-inlier score [3] or attention [87], whereas they need special network design and rely on large amount of training data. In this chapter, we reuse feature extraction network with neglected cost to obtain the class activation map (CAM), which is a good indicator for the foreground and background areas. However, the original CAM is not well calibrated for source and target images, e.g., same part of an object from two images may have different values. Therefore, we propose a staircase function to re-weight pixels of an activation map into four levels: hot spots, object, context and background with decreasing values. With staircase re-weighting, background pixels are unlikely assigned to foreground, thus reducing the background matching.

We combine all proposed modules in a unified optimal transport framework. This is implemented by converting the correlation maximization to optimal transport formulation and incorporating the staircase weights to act as empirical distributions in optimal transport. We summarize the main contributions as follows:

- We model semantic correspondence as an optimal transport problem (SCOT) in a unified framework. The row sum and column sum constraints can be naturally incorporated to suppress many to one matching.

- We propose a staircase function applied on the class activation maps with neglected cost to suppress the background matching.

- The proposed algorithm achieves state-of-the-art performance on four benchmark datasets, especially a 26% relative improvement on the large-scale SPair-71k dataset.

6.2 Proposed Algorithm

In this section, we first introduce preliminary knowledge about optimal transport theory, then we describe how the semantic correspondence problem can be modeled in optimal transport framework, at last, we describe the implementation details about pre- and post-processing.

6.2.1 Preliminary

Optimal transport aims at computing a minimal cost transportation between a source distribution μ_s, and a target distribution μ_t. μ_s and μ_t are defined on probability space $X, Y \in \Omega$, respectively. When a meaningful cost function $c : X \times Y \mapsto \mathbb{R}^+$ is defined, the Kantorovich formulation [151] solves optimal transport by seeking for a probabilistic coupling $\boldsymbol{\pi} \in \mathcal{P}(X \times Y)$:

$$\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi} \in \Pi(\mu_s, \mu_t)}{\operatorname{argmin}} \int_{X \times Y} c(\boldsymbol{x}, \boldsymbol{y}) \boldsymbol{\pi}(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \,, \tag{6.1}$$

where $\Pi(\mu_s, \mu_t) = \{\int_Y \boldsymbol{\pi}(x, y) d\boldsymbol{y} = \mu_s, \int_X \boldsymbol{\pi}(x, y) d\boldsymbol{x} = \mu_t, \boldsymbol{\pi} \geq \mathbf{0}\}$, i.e., $\boldsymbol{\pi}$ is the joint probability measure with marginals μ_s and μ_t.

Here we consider the case when those distributions are discrete empirical distributions, and can be written as

$$\mu_s = \sum_{i=1}^{n_s} p_i^s \delta(x_i) \,, \quad \mu_t = \sum_{i=1}^{n_t} p_i^t \delta(y_i) \,, \tag{6.2}$$

where $\delta(\cdot)$ denotes the Dirac function, n_s and n_t are the number of samples, p_i^s and p_i^t are the probability mass to the i-th sample, belonging to the probability simplex, i.e., $\sum_{i=1}^{n_s} p_i^s = \sum_{i=1}^{n_t} p_i^t = 1$. We define a cost matrix $\mathbf{M}$ with $\mathbf{M}_{ij}$ representing the distance between x_i and y_j. The optimal transport problem is:

$$\begin{aligned} T^* &= \underset{T \in \mathbb{R}_+^{n_s \times n_t}}{\operatorname{argmin}} \sum_{ij} T_{ij} \mathbf{M}_{ij} \\ \text{s.t. } & T \mathbf{1}_{n_t} = \mu_s \,, T^\top \mathbf{1}_{n_s} = \mu_t \,. \end{aligned} \tag{6.3}$$

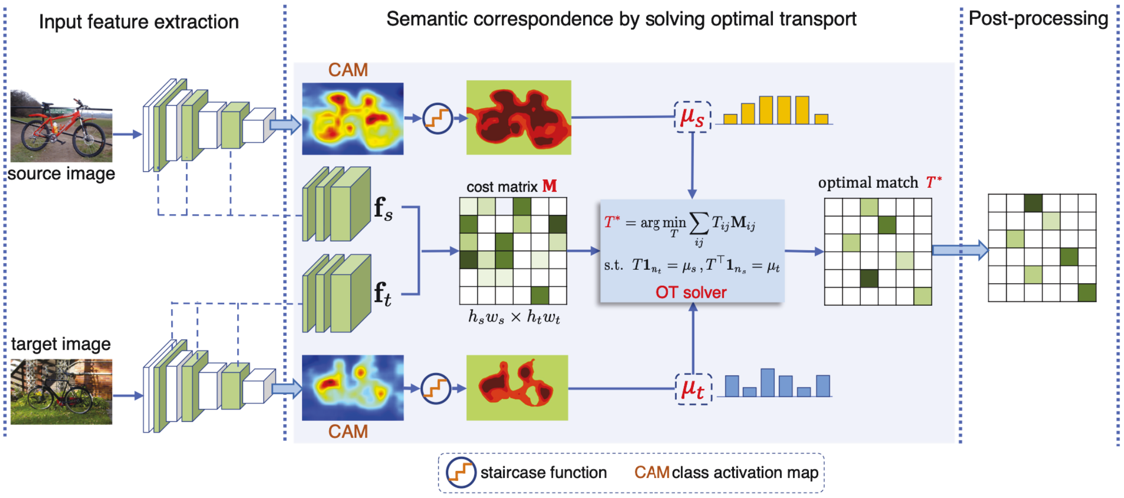


Figure 6.2 : The proposed framework. A pair of images are input to the pre-trained CNN to get the multi-layer feature maps ($\mathbf{f}_s$,$\mathbf{f}_t$) and the class activation maps in a single forward pass. $\mathbf{f}_s$ and $\mathbf{f}_t$ are used to compute a cost matrix $\mathbf{M}$ representing the matching difference. Activation maps undergo a staircase function and are then normalized as the empirical probability distributions μ_s and μ_t. We deal with semantic correspondence by solving optimal transport problem to get the optimal match T^*, which is further post-processed to ensure geometric consistency.

T^* is called the optimal transport plan or transport matrix. T_{ij} denotes the the optimal amount of mass to move from x_i to y_j in order to obtain an overall minimum cost.

6.2.2 Semantic correspondence as an OT problem

Given an input image pair $(\mathbf{I}_s, \mathbf{I}_t)$ containing the same object, the goal of semantic correspondence is to estimate a matrix (e.g., T^* in Figure 6.2) representing the dense matching scores between pixels in two images. A key step in semantic correspondence is to compute the correlation map, which describes the matching similarities between any two locations from different images.

Correlation map A common strategy for computing correlation map is based on matching individual image features using cosine similarity. Given dense feature maps $\mathbf{f}_s \in \mathbb{R}^{h_s \times w_s \times D}$ and $\mathbf{f}_t \in \mathbb{R}^{h_t \times w_t \times D}$ of source and target images extracted from CNNs, the correlation map is computed as:

$$\mathbf{C} = \frac{\mathbf{f}_s \cdot \mathbf{f}_t^\top}{\|\mathbf{f}_s\| \|\mathbf{f}_t\|} \in \mathbb{R}^{h_s \times w_s \times h_t \times w_t}. \tag{6.4}$$

$\mathbf{C}_{ijkl}$ denotes the matching score between the (i, j)-th position in source feature map and (k, l)-th position in target feature map. The best match for (i, j) is computed as $\operatorname{argmax}_{kl} \mathbf{C}_{ijkl}$.

In this process, each of the pairwise matching scores in position (i, j, k, l) is computed individually, without considering any mutual relation or additional constraints. However, since the large intra-class variation and background clutter are ubiquitous in semantic correspondence, this individual strategy often leads to two problems in matching (see Figure 6.1). Firstly, many source positions can be assigned to the same target position due to the individual argmax assignment. This is an undesired property because for the same object it is more reasonable to match each part in source image to the corresponding part in target image, e.g., one-to-one matching. Secondly, foreground object may be assigned to the background due to high feature variation or illumination changes.

Optimal transport problem In this chapter, instead of individual matching strategy, we model this problem from a global perspective. We first introduce a matrix $T \in \mathbb{R}^{h_s w_s \times h_t w_t}$ as the pairwise matching probability from source to target image. Then we resize correlation map $\mathbf{C}$ to the same shape as T and define the ***total correlation*** as $\sum_{ij} T_{ij} \mathbf{C}_{ij}$. Our goal is to maximize the total correlation to get a global optimal matching probability T^*. In order to avoid trivial solutions, we introduce the empirical distribution μ_s and μ_t as the probability of each point in source or target feature map. The values of μ_s and μ_t represent the importance of

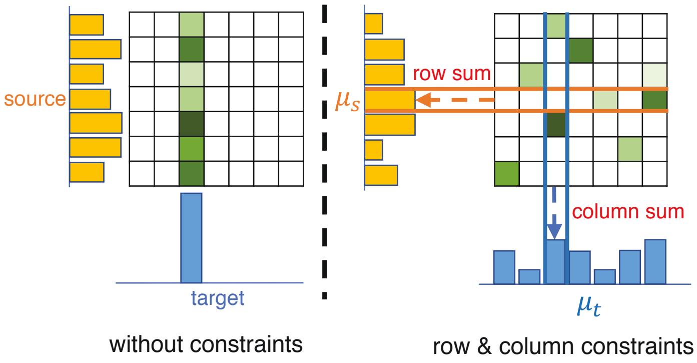


Figure 6.3 : Left: Many source pixels are assigned to one target pixel. Right: The row sum and column sum constraints of the matching matrix suppress the many to one matching.

each point in feature map. Then the marginals of T are constrained to be μ_s and μ_t (i.e., the row sum of T is μ_s and column sum is μ_t). The problem is formulated as:

$$\begin{aligned} T^* = &\underset{T\in\mathbb{R}_+^{h_s w_s \times h_t w_t}}{\operatorname{argmax}} \sum_{ij} T_{ij}\mathbf{C}_{ij} \\ \text{s.t.} \quad &T\mathbf{1}_{h_t w_t} = \mu_s \, , T^\top \mathbf{1}_{h_s w_s} = \mu_t \, . \end{aligned} \tag{6.5}$$

If we define $\mathbf{M} = 1 - \mathbf{C}$ as the cost matrix denoting the matching difference, then Eq. 6.5 is equivalent to minimize the total matching difference:

$$\begin{aligned} T^* = &\underset{T\in\mathbb{R}_+^{h_s w_s \times h_t w_t}}{\operatorname{argmin}} \sum_{ij} T_{ij}\mathbf{M}_{ij} \\ \text{s.t.} \quad &T\mathbf{1}_{h_t w_t} = \mu_s \, , T^\top \mathbf{1}_{h_s w_s} = \mu_t \, , \end{aligned} \tag{6.6}$$

which is a standard optimal transport problem as in Eq. 6.3.

The intuition of modeling semantic correspondence as optimal transport is shown in Figure 6.3. Since the row sum and column sum of matching probability matrix is constrained to be μ_s and μ_t, the many to one matching problem occurring in other cosine similarity based methods [87, 1, 2], are significantly suppressed.

Computation of μ_s and μ_t with staircase re-weighting If we do not have any prior knowledge, then μ_s and μ_t can be set to uniform distributions, indicating same importance of each point in source and target feature maps. Since semantic correspondence suffers from background clutter issue, it is natural to recognize that the foreground object and background should be assigned different importance. Although some previous work [87, 3] showed similar idea, our method is flexible enough to incorporate any kind of prior into the unified optimal transport framework.

We generate the class activation maps [90] for source and target image as the prior information. Since we already have the feature extraction CNNs (detailed in next section), these maps are nearly zero cost due to the same forward pass with feature extraction. Let $\mathbf{f}_L \in \mathbb{R}^{h_L \times w_L \times d_L}$ denote the feature map of the last convolutional layer. It is fed into a Global Average Pooling layer (GAP) followed by a fully connected layer and a softmax layer for classification. The average value of the k_{th} feature map is $s_k = \frac{\sum_{i,j} \mathbf{f}_L(i,j,k)}{h_L \times w_L}$. $W^{fc} \in \mathbb{R}^{d_L \times C}$ denotes the fully connected layer weights, where C is the number of classes. Ignoring the bias term, the input to c_{th} softmax node can be defined as $y_c^{fc} = \sum_{k=0}^{d_L-1} s_k W_{k,c}^{fc}$. The class activation map (CAM) of class c is obtained as follows,

$$A_c = \sum_{k=0}^{d_L-1} \mathbf{f}_L(\cdot, \cdot, k) \cdot W_{k,c}^{fc} \,. \tag{6.7}$$

We choose the class with the highest classification probability and normalize A_c to the range of $[0, 1]$.

The original CAM is not well calibrated for source and target images, e.g., same part of an object from two images may have different values. Therefore, we propose a staircase function to categorize the activation map into four levels according to their values: *hot spots*, *object*, *context* and *background*. Values of each category are adjusted as:

$$A_c(x, y) = \sum_{i=1}^{L} \gamma_i \mathbb{I}(A_c(x, y) > \beta_i) \,, \tag{6.8}$$

Algorithm 4 Optimal transport with sinkhorn algorithm.

```
**Input:** $\mu_s, \mu_t, \mathbf{M}, \epsilon, t_{max}$
Initialize $\mathbf{K} = e^{-\mathbf{M}/\epsilon}$ , $\mathbf{b} \leftarrow \mathbf{1}$ , $t \leftarrow 0$
**while** $t \leq t_{max}$ and not converge **do**
    $\mathbf{a} = \mu_s/(\mathbf{K}\mathbf{b})$
    $\mathbf{b} = \mu_t/(\mathbf{K}^\top \mathbf{a})$
**end while**
**Output:** $T = \text{diag}(\mathbf{a})\mathbf{K}\text{diag}(\mathbf{b})$
```

where $L = 4$ is the number of levels, β_i is the stair height denoting the threshold of i-th level, $\mathbb{I}(\cdot)$ is an indicator function whose value equals 1 only when the condition satisfies, γ_i is the stair width denoting the increased weight from previous level. γ_i and β_i are selected according to the validation set. Now, μ_s and μ_t can be computed as: $\mu_s(x, y) = A_c^s(x, y)/\sum A_c^s(x, y)$, $\mu_t(x, y) = A_c^t(x, y)/\sum A_c^t(x, y)$ and are then flattened to vectors. We call this strategy the *staircase re-weighting.*

Solving OT with Sinkhorn algorithm. Exactly solving Eq. 6.6 with Network Flow solver requires the complexity of $O(n^3)$ (n proportional to $h_s w_s$ and $h_t w_t$). Following [72], we resort to the entropy-regularized optimal transport problem:

$$T^* = \underset{T \in \mathbb{R}_+^{h_s w_s \times h_t w_t}}{\text{argmin}} \sum_{ij} T_{ij} \mathbf{M}_{ij} + \epsilon H(T)$$
$$\text{s.t. } T\mathbf{1}_{h_t w_t} = \mu_s \, , T^\top \mathbf{1}_{h_s w_s} = \mu_t \, , \tag{6.9}$$

where $H(T) = \sum_{ij} T_{ij}(\log T_{ij} - 1)$ is the negative entropic regularization and $\epsilon > 0$ is the regularization parameter. Eq. 6.9 is a convex problem and can be solved using Sinkhorn-Knopp algorithm [137] with the complexity of $O(h_s w_s \times h_t w_t)$. Detailed solution is presented in Algorithm 4. Note that as Algorithm 4 only contains matrix multiplication and exponential operations, it is differentiable and can be computed efficiently. [†]

6.2.3 Pre- and post-processing

Input feature extraction Previous work [88, 3] usually extract feature from the last convolutional layer of deep neural network as matching primitives for semantic correspondence. However, this single layer feature cannot make full use of multi-level representations and fails to deal with ambiguous matching caused by intra-class variations. We follow the good practice of [1] to search and select multi-layer features from all candidate layers of a pre-trained CNN model.

A typical CNN takes an input image and produces a consecutive list of feature maps: $[\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_L]$ with $\mathbf{f}_i \in \mathbb{R}^{h_i \times w_i \times d_i}$. We use the percentage of correct keypoints (PCK) on validation set as a evaluation metric to compare different feature subsets. In order to search the optimal subset of feature maps, we run a variant of beam-search with a limited memory [1]. We maintain a memory containing at most N (beam size) subsets of layers. At each search step, we form the new subsets by adding the candidate layer to each of the current subsets in memory. With all old and new subsets, only the N top performing ones are kept in memory. This process goes on until we reach the maximum number of layers allowed.

After layer search, each image can be represented by a dense spatial feature grid $\mathbf{f} = [\mathbf{f}_{l_1}, \Phi(\mathbf{f}_{l_2}), \ldots, \Phi(\mathbf{f}_{l_k})] \in \mathbb{R}^{h \times w \times D}$, where l_i is the selected layer index and Φ denotes an upsample function. We denote the features of source and target images as $\mathbf{f}_s$ and $\mathbf{f}_t$ respectively.

Post-processing In order to get the geometrically consistent matching, we employ the regularized Hough matching (RHM) [1] as the post-processing step.

[†] In this work, we only use pre-trained CNN model. We implement a GPU version of Algorithm 4 for fast computation.

Let us assume $\mathbf{p}_s$ and $\mathbf{p}_t$ the position grids of feature maps $\mathbf{f}_s$ and $\mathbf{f}_t$. $R_s = (\mathbf{f}_s, \mathbf{p_s})$ and $R_t = (\mathbf{f}_t, \mathbf{p_t})$ are the coupled feature-position sets with r,r' being their elements. For the sake of simplicity, we denote $\mathcal{D}$ for two sets and m for a match: $\mathcal{D} = (R_s, R_t), m = (r, r')$ in $R_s \times R_t$. The matching confidence for m is denoted as $p(m|\mathcal{D})$. Since the source and target images contain the same object, we assume the common object can be located with *offset* x lying in a Hough space $\mathcal{X}$. The matching confidence can be calculated as follows:

$$p(m|\mathcal{D}) = p(m_a) \sum_{x \in \mathcal{X}} p(m_g|x) p(x|\mathcal{D}) \,, \tag{6.10}$$

$$p(x|\mathcal{D}) \propto \sum_{m} p(m_a) p(m_g|x) \,. \tag{6.11}$$

Here $p(m_a)$ is the appearance score, $p(m_g|x)$ is the geometric score given an offset x, $p(x|\mathcal{D})$ is the geometry prior computed by aggregating individual votes into the Hough space scores.

In this work, we set $p(m_a) = T^*$. For $p(m_g|x)$, we estimate it by comparing $\mathbf{p}_s(i,j) - \mathbf{p}_t(i,j)$ to the given offset x. The two-dimensional offset bins is constructed and a Gaussian mask is centered on offset x to re-weight the values.

After we obtain the matching confidence $p(m|D)$, it is easy to transfer any keypoint from a source image to the target image. Given a keypoint x_p in a source image, we first compute the neighborhood pixels $\mathcal{N}(x_p)$ covered within the receptive fields of the feature map, and we compute the displacement between x_p and middle of the receptive fields, *i.e.*, $\{d(x_q)\}_{x_q \in \mathcal{N}(x_p)}$. Let y_q denotes the target point for x_q computed from $p(m|D)$ by nearest neighbor assignment. The corresponding keypoint for x_p is the average of $\{y_q + d(x_q)\}_{x_q \in \mathcal{N}(x_p)}$.

6.3 Experiments

In this section we describe our benchmarks and evaluation metric, give implementation details, and compare our method to baselines and the state-of-the-art.

6.3.1 Benchmarks and evaluation metric

SPair-71k [152]. Because of the annotation cost, previous datasets are relatively small and do not show much variability. The newly-released SPair-71k dataset includes 70,958 image pairs with various viewpoint and scale changes, which is a reliable testbed for studying real problems of semantic correspondence.

TSS [147], **PF-PASCAL** [153], and **PF-WILLOW** [82]. TSS contains 400 image pairs divided into three groups: FG3DCar [154], JODS [146], and PASCAL [155]. PF-PASCAL contains 1,351 image pairs from the 20 object categories of the PASCAL VOC [156] dataset. PF-WILLOW contains 900 image pairs of 4 object categories. For a fair comparison, we follow the settings of previous work [86, 84, 9, 1, 3] to evaluate our model.

Evaluation metric. We employ the commonly-used metric of percentage of correct keypoints (PCK). It calculates the number of predicted keypoints that is correct under a fixed threshold. Once we have the predicted keypoint $\mathbf{k}_{pr}$ and the ground-truth keypoint $\mathbf{k}_{gt}$, if the following condition satisfies:

$$d(\mathbf{k}_{pr}, \mathbf{k}_{gt}) \leq \alpha_\tau \cdot \max(w_\tau, h_\tau), \tag{6.12}$$

then the prediction is correct. Here, $d(\cdot, \cdot)$ is the Euclidean distance, w_τ and h_τ represent the width and height (image level or bounding box level) which are determined according to the criterion $\tau \in \{\text{img}, \text{bbox}\}$, where α_τ is a fixed threshold (e.g., $\alpha_\tau = 0.1$). PCKs of all image pairs are averaged to get the final PCK. Following [1], we evaluate PF-PASCAL with α_{img}, PF-WILLOW and SPair-71k with the more stringent criterion α_{bbox}. For TSS, we follow [147, 84] to compute the PCK over a dense set of keypoints.

6.3.2 Implementation details

In this section, two network structures are utilized as the backbone for feature and activation map extraction: ResNet50 and ResNet101 [7] pre-trained on ImageNet [157]. No fine-tuning is performed in any manner in our algorithm.

To select multiple-layer features, we run the search algorithm proposed in [1] with the proposed optimal transport matching on validation set. For SPair-71k, the best layer subsets are $(0, 11, 12, 13)$ with ResNet-50 and $(0, 19, 27, 28, 29, 30)$ with ResNet-101. For PF-PASCAL and PF-WILLOW, the best layer subsets are $(2, 22, 24, 25, 27, 28, 29)$ with ResNet-101. The optimal layers are different from [1] since we consider the total correlations rather than individual feature matching.

In Eq. 6.8, we set $\boldsymbol{\gamma} = [0.5, 0.3, 0.1, 0.1]$ and $\boldsymbol{\beta} = [0.0, 0.4, 0.5, 0.6]$ according to the PCK of the validation set. In Algorithm 4, we set $\epsilon = 0.05$ and $t_{max} = 50$.

6.3.3 Evaluation results on SPair-71k

Comparisons to state-of-the-art First, we compare per-class PCK on the SPair-71k dataset with state-of-the-art methods in Table 6.1. The overall PCK of the proposed algorithm outperforms the state-of-the-art [1] by 7.4 (relative 26%), which is a huge improvement. And for all classes, our algorithm surpasses [1] by a large margin. Among all candidate algorithms, our algorithm achieves the best PCK on 16 out of 18 classes. This proves the effectiveness and robustness of our optimal transport algorithm in finding global optimal matching.

To better understand the performance of our algorithm under complex conditions, we report the results according to different variation factors with various difficulty levels in Table 6.2. SPair-71k dataset contains diverse variations in view-point, scale, truncation and occlusion, which is a reliable testbed to study the problem of semantic correspondence. The results clearly show that the proposed algorithm out-

Table 6.1 : Results on SPair-71k (α_{bbox}=0.1). All models in this table use ResNet101 as the backbone. For the authors' original models, [88, 87] were trained on PASCAL-VOC with self-supervision, [3, 2] were trained on PF-PASCAL with weal-supervision. For SPair-71k-finetuned models, we follow HPF [1] to further finetune the original models on SPair-71k dataset. For SPair-71k validation models, [1] and our method only utilize the validation split for hyperparameter tuning. The best performances are shown in bold.

Methods		aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dog	horse	moto	person	plant	sheep	train	tv	all
Authors' original models	CNNGeo [88]	21.3	15.1	34.6	12.8	31.2	26.3	24.0	30.6	11.6	24.3	20.4	12.2	19.7	15.6	14.3	9.6	28.5	28.8	18.1
	A2Net [87]	20.8	17.1	37.4	13.9	33.6	29.4	26.5	34.9	12.0	26.5	22.5	13.3	21.3	20.0	16.9	11.5	28.9	31.6	20.1
	WeakAlign [3]	23.4	17.0	41.6	14.6	37.6	28.1	26.6	32.6	12.6	27.9	23.0	13.6	21.3	22.2	17.9	10.9	31.5	34.8	21.1
	NC-Net [2]	24.0	16.0	45.0	13.7	35.7	25.9	19.0	50.4	14.3	32.6	27.4	19.2	21.7	20.3	20.4	13.6	33.6	40.4	26.4
SPair-71k finetuned models	CNNGeo [88]	23.4	16.7	40.2	14.3	36.4	27.7	26.0	32.7	12.7	27.4	22.8	13.7	20.9	21.0	17.5	10.2	30.8	34.1	20.6
	A2Net [87]	22.6	18.5	42.0	16.4	37.9	**30.8**	26.5	35.6	13.3	29.6	24.3	16.0	21.6	22.8	20.5	13.5	31.4	36.5	22.3
	WeakAlign [3]	22.2	17.6	41.9	15.1	38.1	27.4	**27.2**	31.8	12.8	26.8	22.6	14.2	20.0	22.2	17.9	10.4	32.2	35.1	20.9
	NC-Net [2]	17.9	12.2	32.1	11.7	29.0	19.9	16.1	39.2	9.9	23.9	18.8	15.7	17.4	15.9	14.8	9.6	24.2	31.1	20.1
SPair-71k validation	HPF [1]	25.2	18.9	52.1	15.7	38.0	22.8	19.1	52.9	17.9	33.0	32.8	20.6	24.4	27.9	21.1	15.9	31.5	35.6	28.2
	Ours	**34.9**	**20.7**	**63.8**	**21.1**	**43.5**	27.3	21.3	**63.1**	**20.0**	**42.9**	**42.5**	**31.1**	**29.8**	**35.0**	**27.7**	**24.4**	**48.4**	**40.8**	**35.6**

Table 6.2 : PCK analysis by variation factors on SPair-71k ($\alpha_{\text{bbox}} = 0.1$). The variation factors include view-point, scale, truncation, and occlusion with various difficulty levels. All models in this table use ResNet101 as the backbone.

Methods		View-point			Scale			Truncation				Occlusion				All
		easy	medi	hard	easy	medi	hard	none	src	tgt	both	none	src	tgt	both	
Identity mapping		7.3	3.7	2.6	7.0	4.3	3.3	6.5	4.8	3.5	5.0	6.1	4.0	5.1	4.6	5.6
Authors' original models	CNNGeo [88]	25.2	10.7	5.9	22.3	16.1	8.5	21.1	12.7	15.6	13.9	20.0	14.9	14.3	12.4	18.1
	A2Net [87]	27.5	12.4	6.9	24.1	18.5	10.3	22.9	15.2	17.6	15.7	22.3	16.5	15.2	14.5	20.1
	WeakAlign [3]	29.4	12.2	6.9	25.4	19.4	10.3	24.1	16.0	18.5	15.7	23.4	16.7	16.7	14.8	21.1
	NC-Net [2]	34.0	18.6	12.8	31.7	23.8	14.2	29.1	22.9	23.4	21.0	29.0	21.1	21.8	19.6	26.4
SPair-71k finetuned models	CNNGeo [88]	28.8	12.0	6.4	24.8	18.7	10.6	23.7	15.5	17.9	15.3	22.9	16.1	16.4	14.4	20.6
	A2Net [87]	30.9	13.3	7.4	26.1	21.1	12.4	25.0	17.4	20.5	17.6	24.6	18.6	17.2	16.4	22.3
	WeakAlign [3]	29.3	11.9	7.0	25.1	19.1	11.0	24.0	15.8	18.4	15.6	23.3	16.1	16.4	15.7	20.9
	NC-Net [2]	26.1	13.5	10.1	24.7	17.5	9.9	22.2	17.1	17.5	16.8	22.0	16.3	16.3	15.2	20.1
SPair-71k validation	HPF [1]	35.6	20.3	15.5	33.0	26.1	15.8	31.0	24.6	24.0	23.7	30.8	23.5	22.8	21.8	28.2
	Ours	**42.7**	**28.0**	**23.9**	**41.1**	**33.7**	**21.4**	**39.0**	**32.4**	**30.0**	**30.0**	**39.0**	**30.3**	**28.1**	**26.0**	**35.6**

performs all other methods by a large margin in all conditions, which demonstrates the high stability of the proposed algorithm.

Ablation studies on feature matching To verify the effect of global feature matching in optimal transport, we compare optimal transport with individual feature matching (i.e, cosine) on SPair-71k. We first introduce the baselines. Here, "**Cos-NN**" denotes the cosine matching scores (Eq. 6.4) followed by nearest neighbor assignment (NN). "**Cos-RHM**" denotes the cosine matching scores followed by regularized Hough matching (RHM), which is equivalent to the HPF algorithm in [1] [‡]. Similarly, "**OT-NN**" and "**OT-RHM**" denote matching scores computed by optimal transport (Eq. 6.9) *without class activation maps* (i.e., μ_s and μ_t are set to uniform distribution) followed by corresponding post-processing. Finally, "**OT-RHM-CAM**" denotes the baseline using the original class activation maps, while "**OT-RHM-Stair**" denotes our model with staircase re-weighting on class activation maps, which is our ultimate model.

The results are shown in Table 6.3. It can be seen that in all settings (various backbones and geometric post-processing), the proposed optimal transport solution beats the corresponding baseline by a large margin. In order to study the *many to one matching* issue shown in Figure 6.1a, we calculated the average number of unique points on target maps assigned to source. As shown in Table 6.3, optimal transport has about twice individual matches as many as cosine methods, this agrees with our motivation that global feature matching can significantly suppress the *many to one matching.* We further observe that RHM can increase the number of target matches for both cosine and optimal transport. However, the PCK of Cos-RHM even drops compared with Cos-NN, while OT-RHM continues to increase over OT-NN. We conjecture that too many duplicated assignments of cosine matching hinders

[‡] For fair comparison, we use multiple-layer features selected in Sec 6.3.2 instead of [1].

Table 6.3 : PCK results ($\alpha_{\text{bbox}} = 0.1$) on SPair-71k dataset with feature matching computed by cosine (Cos) and optimal transport (OT). "src pts" denotes the average number of points from source feature maps. "trg matches" denotes the average number of unique points on target maps assigned to the source points.

Backbone	Methods	src pts	trg matches	PCK
ResNet50	Cos-NN	4099	558	28.0
	OT-NN	4099	**1184**	**29.4**
	Cos-RHM	4099	783	26.6
	OT-RHM	4099	**1322**	**31.3**
ResNet101	Cos-NN	4099	445	30.6
	OT-NN	4099	**1062**	**33.7**
	Cos-RHM	4099	701	27.8
	OT-RHM	4099	**1261**	**34.8**

Table 6.4 : Ablation study of staircase re-weighting on SPair-71k dataset. PCK results with $\alpha_{\text{bbox}} = 0.1$ are reported.

Methods	ResNet50	ResNet101
OT-NN	29.4	33.2
OT-NN-CAM	29.6	32.6
OT-NN-Stair	**30.2**	**34.2**
OT-RHM	31.3	34.8
OT-RHM-CAM	31.4	34.5
OT-RHM-Stair	**32.1**	**35.6**

the effectiveness of RHM for better geometric adjustment. We also notice that ResNet101 has a smaller number of target matches than ResNet50. The reason is that the deeper network has a larger receptive field in deeper layers that makes features of these layers less distinguishable.

Ablation studies on staircase re-weighting. We then investigate the effect of staircase re-weighting under different backbones and geometric post-processing. From Table 6.4, we can see that original CAM has little or no improvement compared to the baselines while our staircase re-weighting enjoys at least 0.8% PCK increase. Considering the staircase re-weighting strategy shares the same CNN forward pass with feature extraction and the extra cost is nearly free, this is a promising improvement. We believe this strategy can perform better with more accurate class activation maps. We leave this for future study.

Visualization. We show the qualitative results in Figure 6.4. We warp the source images to align with the corresponding target images. For HPF [1], NC-net [2], and our method, we first use the source keypoints and the predicted target keypoints to estimate the thin-plate spline (TPS) parameters, then apply TPS transformation on the source image. For A2Net [1] and WeakAlign [3], they are global alignment methods that directly predict the global transformation parameters from the CNN models. We show the results of image pairs with large intra-class, scale, and viewpoint changes. Our method performs better in complex conditions due to our global matching and background suppressing strategies.

6.3.4 TSS, PF-PASCAL, and PF-WILLOW

Table 6.5 shows the evaluation results on TSS dataset. The proposed method outperforms previous methods on one of the three groups of the TSS dataset and the average performance over three groups on the TSS dataset sets up a new state of the art.

Table 6.6 summarizes comparisons to state-of-the-art methods on PF-PASCAL and PF-WILLOW. Following [1], we use the backbone of FCN [10] pre-trained with

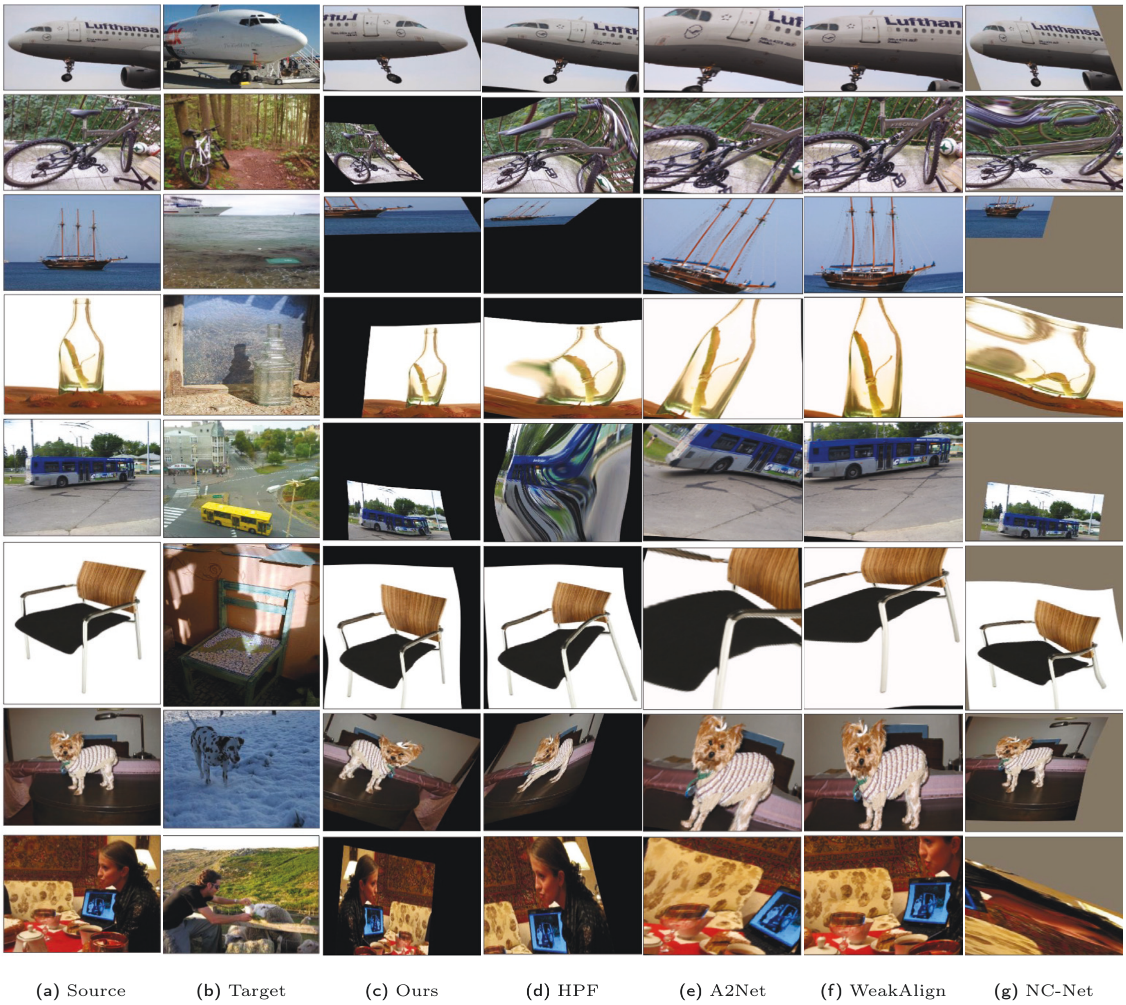


Figure 6.4 : Qualitative results on SPari-71k. The source images are warped to align with target images using correspondences. For HPF [1], NC-net [2], and our method, we first use the source keypoints and the predicted target keypoitns to estimate the thin-plate spline (TPS) parameters, then apply TPS transformation on the source image. For A2Net [1] and WeakAlign [3], they are global alignment methods that directly predict the global transformation parameters from the CNN models. We show image pairs with large intra-class, scale, and view-point changes. Our method performs better in complex conditions due to our global matching and background suppressing strategies.

Table 6.5 : Evaluation results on TSS dataset. Subscripts of the method names indicate backbone networks used. We report the PCK scores with $\alpha = 0.05$ and the best results are in bold.

Methods	FG3D.	JODS	PASC.	Avg.
CNNGeo$_{\text{res101}}$ [88]	90.1	76.4	56.3	74.3
DCTM$_{\text{CAT-FCSS}}$ [158]	89.1	72.1	61.0	74.0
Weakalign$_{\text{res101}}$ [3]	90.3	76.4	56.5	74.4
RTNs$_{\text{res101}}$ [150]	90.1	78.2	**63.3**	77.2
NC-Net$_{\text{res101}}$ [2]	94.5	81.4	57.1	77.7
DCCNet$_{\text{res101}}$ [84]	93.5	**82.6**	57.6	77.9
HPF$_{\text{res101}}$ [1]	93.6	79.7	57.3	76.9
Ours$_{\text{res101}}$	**95.3**	81.3	57.7	**78.1**

PASCAL VOC 2012 [156] [§]. Different levels of supervisory signals are used in the deep network models, such as self-supervision [88, 87], weak-supervision [158, 3, 2, 84, 150], keypoints [86, 89] and masks [9]. In the contrary, HPF [1] and our method only use the pre-trained ImageNet models and the validation set. Table 6.6 shows that the proposed method achieves the state-of-the-art results on both benchmarks with various thresholds α. It need to be further noticed that when α becomes smaller (stricter criterion), our method gains larger advantage over others. This indicates that our method generates more accurate keypoint predictions, so it can perform better with small threshold.

6.4 Conclusion

We propose to model semantic correspondence as an optimal transport problem (SCOT). We solve semantic correspondence by maximizing the total correlations

[§] For this network, we directly extract the max-aggregated class masks as class activation map. Image-level annotations are not used.

Table 6.6 : Evaluation results on PF-PASCAL and PF-WILLOW. Subscripts of the method names indicate backbone networks used. Different levels of supervision are used, such as self-supervision [88, 87], weak-supervision [158, 3, 2, 84, 150], keypoints [86, 89] and masks [9]. HPF [1] and our method only use pre-trained models and the validation set. The best performances are shown in bold. We borrow the results of [86, 82, 158, 88, 3] from [150].

Methods	PF-PASCAL (α_{img})			PF-WILLOW (α_{bbox})		
	0.05	0.1	0.15	0.05	0.1	0.15
PF$_{\mathrm{HOG}}$ [82]	31.4	62.5	79.5	28.4	56.8	68.2
CNNGeo$_{\mathrm{res101}}$ [88]	41.0	69.5	80.4	36.9	69.2	77.8
A2Net$_{\mathrm{res101}}$ [87]	42.8	70.8	83.3	36.3	68.8	84.4
DCTM$_{\mathrm{CAT\text{-}FCSS}}$ [158]	34.2	69.6	80.2	38.1	61.0	72.1
Weakalign$_{\mathrm{res101}}$ [3]	49.0	74.8	84.0	37.0	70.2	79.9
NC-Net$_{\mathrm{res101}}$ [2]	54.3	78.9	86.0	33.8	67.0	83.7
DCCNet$_{\mathrm{res101}}$ [84]	-	82.3	-	43.6	73.8	86.5
RTNs$_{\mathrm{res101}}$ [150]	55.2	75.9	85.2	41.3	71.9	86.2
SCNet$_{\mathrm{VGG16}}$ [86]	36.2	72.2	82.0	38.6	70.4	85.3
NN-Cyc$_{\mathrm{res101}}$ [89]	55.1	85.7	94.7	40.5	72.5	86.9
SFNet$_{\mathrm{res101}}$ [9]	-	78.7	-	-	74.0	-
HPF$_{\mathrm{res101}}$ [1]	60.1	84.8	92.7	45.9	74.4	85.6
HPF$_{\mathrm{res101\text{-}FCN}}$ [1]	63.5	88.3	**95.4**	48.6	76.3	88.2
Ours$_{\mathrm{res101}}$	63.1	85.4	92.7	47.8	76.0	87.1
Ours$_{\mathrm{res101\text{-}FCN}}$	**67.3**	**88.8**	**95.4**	**50.7**	**78.1**	**89.1**

between pixels in two images, which is equivalent to the standard optimal transport problem. We then apply a staircase function on the class activation maps generated from feature extraction CNNs with neglected extra cost to re-weight the importance of foreground and background pixels. These re-weighted maps are normalized to serve as prior information for empirical distributions in optimal transport. The ablation studies clearly demonstrate the effectiveness of each component. And SCOT outperforms state-of-the-art on standard benchmarks by a large margin.

Chapter 7

Conclusion and Future Work

This thesis focuses on the learning algorithms with limited labeled data and applies them to solve two problems: few-shot learning and object matching.

For few-shot learning, I focus on two settings: few-shot image classification and online learning with imbalanced streaming data. In the formal setting, a *Transductive Propagation Network* (TPN) is proposed to learn to propagate labels from labeled instances to unlabeled test instances, thus alleviating the low-data issue. In the latter setting, an adaptive sparse confidence-weighted (ASCW) algorithm is proposed to explore the feature correlation and maintain multiple learners with different costs to address the imbalanced issue. For object matching, due to the annotation difficulty, a good strategy is to explore high-confidence matching pairs from unlabeled objects. Based on this strategy, two algorithms are proposed. First, a squared-loss mutual information (SMI) is applied to utilize a small number of paired samples and the abundant unpaired ones. Second, the specific object matching problem, semantic correspondence, is solved in a unified optimal transport framework to address the many to one matching and background matching issues.

From the above solutions for limited labeled data, some good practice can be summarized: (1) an expected model should be parameter-efficient to cope with the overfitting issue caused by the limited labeled data. (2) in order to combine with powerful models such as deep neural networks, the expected model should induce simple and explicit gradient computation for efficient model training. In this thesis, closed-form algorithms and Sinkhorn algorithm are two practical instantialization that is consistent with these two practices.

Most of the proposed algorithms deal with limited labeled data learning and generalization in a single domain (dataset). A promising future direction is to investigate the generalization across multiple domains for more realistic applications. For multiple domains, the potential model should have diverse schemes to deal with the heterogeneous tasks from multiple domains. A challenge is how to encourage positive transfer across similar domains and prevent negative interference across different domains. However, the proposed algorithms can be a good starting point to deal with multi-domain problems, *e.g.*, maintaining several TPN network or multi-graph matching using Sinkhorn algorithm.

Bibliography

[1] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *ICCV*, 2019.

[2] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NeurIPS*, 2018.

[3] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *CVPR*, 2018.

[4] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Devon Hjelm, and Aaron Courville. Mutual information neural estimation. In *ICML*, 2018.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[8] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *CVPR*, 2020.

[9] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. SFNet: Learning Object-aware Semantic Correspondence. In *CVPR*, 2019.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[12] David Kirk et al. Nvidia cuda software and gpu parallel computing architecture. In *ISMM*, volume 7, pages 103–104, 2007.

[13] Larry E Maddux, Raymond J Drake Jr, Louis Anthony VanHoutin, Vincent R Long, and Cortney Warmouth. Tpu/foam jaw pad, June 19 2012. US Patent 8,201,269.

[14] Amy Zhao, Guha Balakrishnan, Fredo Durand, John V Guttag, and Adrian V Dalca. Data augmentation using learned transformations for one-shot medical image segmentation. In *CVPR*, 2019.

[15] Huy V Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. In *CVPR*, 2019.

[16] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[17] YM Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2019.

[18] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020.

[19] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[21] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[22] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[23] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017.

[24] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[25] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *ICLR*, 2017.

[26] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[28] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *ICLR*, 2018.

[29] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*, 2019.

[30] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.

[31] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[32] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019.

[33] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[34] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, 2004.

[35] Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook.* PhD thesis, Technische Universität München, 1987.

[36] Sebastian Thrun and Lorien Pratt. *Learning to learn.* Springer Science & Business Media, 2012.

[37] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[38] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10657–10665, 2019.

[39] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, 2016.

[40] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

[41] Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[42] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[43] Fei Wang and Changshui Zhang. Label propagation through linear neighborhoods. In *ICML*, 2006.

[44] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In *NIPS*, 2013.

[45] Yanwei Fu, Timothy M Hospedales, Tao Xiang, and Shaogang Gong. Transductive multi-view zero-shot learning. *IEEE TPAMI*, 37(11):2332–2345, 2015.

[46] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. *Technical Report, Carnegie Mellon University*, 2002.

[47] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[48] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *JMLR*, 7(Mar):551–585, 2006.

[49] Koby Crammer, Mark Dredze, and Fernando Pereira. Exact convex confidence-weighted learning. In *NIPS*, 2009.

[50] Jialei Wang, Peilin Zhao, and Steven CH Hoi. Exact soft confidence-weighted learning. In *ICML*, 2012.

[51] Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. In *NIPS*, 2009.

[52] Justin Ma, Alex Kulesza, Mark Dredze, Koby Crammer, Lawrence Saul, and Fernando Pereira. Exploiting feature covariance in high-dimensional online learning. In *AISTATS*, 2010.

[53] Xuanyi Dong, Yan Yan, Mingkui Tan, Yi Yang, and Ivor W Tsang. Late fusion via subspace search with consistency preservation. *IEEE TIP*, 28(1):518–528, 2019.

[54] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, 2003.

[55] Jialei Wang, Peilin Zhao, and Steven CH Hoi. Cost-sensitive online classification. *IEEE TKDE*, 26(10):2425–2438, 2014.

[56] Peilin Zhao and Steven CH Hoi. Cost-sensitive online active learning with application to malicious url detection. In *ACM SIGKDD*, 2013.

[57] Chao Han, Yun-Kun Tan, Jin-Hui Zhu, Yong Guo, Jian Chen, and Qing-Yao Wu. Online feature selection of class imbalance via pa algorithm. *Journal of Computer Science and Technology*, 31(4):673–682, 2016.

[58] Peilin Zhao, Yifan Zhang, Min Wu, Steven CH Hoi, Mingkui Tan, and Junzhou Huang. Adaptive cost-sensitive online classification. *IEEE TKDE*, 2018.

[59] Xindong Wu, Kui Yu, Wei Ding, Hao Wang, and Xingquan Zhu. Online feature selection with streaming features. *IEEE TPAMI*, 35(5):1178–1192, 2013.

[60] Peng Zhou, Xuegang Hu, Peipei Li, and Xindong Wu. Online feature selection for high-dimensional class-imbalanced data. *Knowledge-Based Systems*, 136:187–199, 2017.

[61] Kui Yu, Xindong Wu, Wei Ding, and Jian Pei. Towards scalable and accurate online feature selection for big data. In *ICDM*, 2014.

[62] Jialei Wang, Peilin Zhao, Steven CH Hoi, and Rong Jin. Online feature selection and its applications. *IEEE TKDE*, 26(3):698–710, 2014.

[63] Yue Wu, Steven CH Hoi, Tao Mei, and Nenghai Yu. Large-scale online feature selection for ultra-high dimensional sparse data. *ACM TKDD*, 11(4):48, 2017.

[64] Mingkui Tan, Yan Yan, Li Wang, Anton Van Den Hengel, Ivor W Tsang, and Qinfeng (Javen) Shi. Learning sparse confidence-weighted classifier on very high dimensional data. In *AAAI*, 2016.

[65] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[66] Taiji Suzuki, Masashi Sugiyama, and Toshiyuki Tanaka. Mutual information approximation via maximum likelihood estimation of density ratio. In *ISIT*, 2009.

[67] Taiji. Suzuki, Masashi. Sugiyama, Takafumi. Kanamori, and Jun. Sese. Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, 10(S52), 2009.

[68] Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aaron van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. *NeurIPS*, 2019.

[69] Facundo Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487, 2011.

[70] Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *ICML*, 2016.

[71] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[72] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013.

[73] Yuguang Yan, Wen Li, Hanrui Wu, Huaqing Min, Mingkui Tan, and Qingyao Wu. Semi-supervised optimal transport for heterogeneous domain adaptation. In *IJCAI*, 2018.

[74] N. Quadrianto, A.J. Smola, L. Song, and T. Tuytelaars. Kernelized sorting. *IEEE TPAMI*, 32:1809–1821, 2010.

[75] Nemanja Djuric, Mihajlo Grbovic, and Slobodan Vucetic. Convex kernelized sorting. In *AAAI*, 2012.

[76] A. Gretton, O. Bousquet, Alex. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *ALT*, 2005.

[77] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[78] N Danal. Histgram of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[79] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[80] Yehezkel Lamdan, Jacob T Schwartz, and Haim J Wolfson. Object recognition by affine invariant matching. In *CVPR*, 1988.

[81] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*, 2015.

[82] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *CVPR*, 2016.

[83] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *NIPS*, 2016.

[84] Shuaiyi Huang, Qiuyue Wang, Songyang Zhang, Shipeng Yan, and Xuming He. Dynamic context correspondence network for semantic alignment. In *ICCV*, 2019.

[85] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *NIPS*, 2014.

[86] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *ICCV*, 2017.

[87] Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. In *ECCV*, 2018.

[88] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017.

[89] Zakaria Laskar, Hamed Rezazadegan Tavakoli, and Juho Kannala. Semantic matching by weakly supervised 2d point set registration. In *WACV*, 2019.

[90] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[91] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018.

[92] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[93] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE TPAMI*, 39(9):1853–1865, 2016.

[94] Zhengyu Su, Yalin Wang, Rui Shi, Wei Zeng, Jian Sun, Feng Luo, and Xianfeng Gu. Optimal mass transport for shape matching and comparison. *IEEE TPAMI*, 37(11):2246–2259, 2015.

[95] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

[96] Charlotte Bunne, David Alvarez-Melis, Andreas Krause, and Stefanie Jegelka. Learning generative models across incomparable spaces. In *ICML*, 2019.

[97] Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced wasserstein distance and its use for gans. In *CVPR*, 2019.

[98] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *CVPR*, 2019.

[99] Hongteng Xu, Dixin Luo, and Lawrence Carin. Scalable gromov-wasserstein learning for graph partitioning and matching. *arXiv preprint arXiv:1905.07645*, 2019.

[100] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-wasserstein learning for graph matching and node embedding. In *ICML*, 2019.

[101] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Conference of the Cognitive Science Society*, volume 33, 2011.

[102] Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *ICML*, 2018.

[103] Zhongwen Xu, Linchao Zhu, and Yi Yang. Few-shot object recognition from machine-labeled web images. In *CVPR*, 2017.

[104] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018.

[105] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014.

[106] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018.

[107] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. American Mathematical Soc., 1997.

[108] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, 2016.

[109] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *NIPS*, 2004.

[110] Masashi Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *JMLR*, 8:1027–1061, 2007.

[111] De-Ming Liang and Yu-Feng Li. Lightweight label propagation for large-scale network data. In *IJCAI*, 2018.

[112] Yasuhiro Fujiwara and Go Irie. Efficient label propagation. In *ICML*, 2014.

[113] Boris N Oreshkin, Alexandre Lacoste, and Pau Rodriguez. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018.

[114] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[115] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *ICML*, 2018.

[116] Matthias Bauer, Rojas-Carulla Mateo, Jakub Bartłomiej Świątkowski, Bernhard Schölkopf, and Richard E Turner. Discriminative k-shot learning using probabilistic models. *arXiv preprint arXiv:1706.00326*, 2017.

[117] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019.

[118] Aditya Pal and Deepayan Chakrabarti. Label propagation with neural networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1671–1674, 2018.

[119] Barbara Caroline Benato, Jancarlo Ferreira Gomes, Alexandru Cristian Telea, and Alexandre Xavier Falcão. Semi-supervised deep learning based on label propagation in a 2d embedded space. *arXiv preprint arXiv:2008.00558*, 2020.

[120] Hanna Sumita, Yasushi Kawase, Sumio Fujita, Takuro Fukunaga, and RIKEN AIP Center. Online optimization of video-ad allocation. In *IJCAI*, 2017.

[121] Yan Yan, Tianbao Yang, Yi Yang, and Jianhui Chen. A framework of online learning with imbalanced streaming data. In *AAAI*, 2017.

[122] Justin Ma, Lawrence K Saul, Stefan Savage, and Geoffrey M Voelker. Identifying suspicious urls: an application of large-scale online learning. In *ICML*, 2009.

[123] Mingkui Tan, Li Wang, and Ivor W Tsang. Learning sparse svm for feature selection on very high dimensional datasets. In *ICML*, 2010.

[124] Kenneth O Kortanek and Hoon No. A central cutting plane algorithm for convex semi-infinite programming problems. *SIAM Journal on optimization*, 3(4):901–918, 1993.

[125] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *ICML*, 2014.

[126] Andre Filipe Torres Martins, Noah Smith, Eric Xing, Pedro Aguiar, and Mario Figueiredo. Online learning of structured predictors with multiple kernels. In *AISTATS*, 2011.

[127] Doyen Sahoo, Steven Hoi, and Peilin Zhao. Cost sensitive online multiple kernel classification. In *ACML*, 2016.

[128] Mingkui Tan, Ivor W Tsang, and Li Wang. Towards ultrahigh dimensional feature selection for big data. *JMLR*, 15(1):1371–1429, 2014.

[129] Guo-Xun Yuan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. A comparison of optimization methods and software for large-scale l1-regularized linear classification. *JMLR*, 11(Nov):3183–3234, 2010.

[130] Steven CH Hoi, Jialei Wang, and Peilin Zhao. Libol: A library for online learning algorithms. *JMLR*, 15(1):495–499, 2014.

[131] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., Hoboken, NJ, USA, 2nd edition, 2006.

[132] Taiji Suzuki and Masashi Sugiyama. Sufficient dimension reduction via squared-loss mutual information estimation. In *AISTATS*, 2010.

[133] Makoto Yamada and Masashi Sugiyama. Dependence minimizing regression with model selection for non-linear causal inference under non-gaussian noise. In *AAAI*, 2010.

[134] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Infovae: Balancing learning and inference in variational autoencoders. In *AAAI*, 2019.

[135] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[136] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[137] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *Proceedings of the American Mathematical Society*, 45(2):195–198, 1974.

[138] Bernhard Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019.

[139] Makoto Yamada, Leonid Sigal, Michalis Raptis, Machiko Toyoda, Yi Chang, and Masashi Sugiyama. Cross-domain matching with squared-loss mutual information. *IEEE TPAMI*, 37(9):1764–1776, 2015.

[140] Makoto Yamada and Masashi Sugiyama. Cross-domain object matching with model selection. In *AISTATS*, 2011.

[141] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[142] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Gert R Lanckriet, and Bernhard Schölkopf. Kernel choice and classifiability for rkhs embeddings of probability distributions. In *NIPS*, 2009.

[143] R'emi Flamary and Nicolas Courty. Pot python optimal transport library, 2017.

[144] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[145] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011.

[146] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *CVPR*, 2013.

[147] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *CVPR*, 2016.

[148] Kevin Dale, Micah K Johnson, Kalyan Sunkavalli, Wojciech Matusik, and Hanspeter Pfister. Image restoration using online photo collections. In *ICCV*, 2009.

[149] Richard Szeliski et al. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2007.

[150] Seungryong Kim, Stephen Lin, SANG RYUL JEON, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer networks for semantic correspondence. In *NeurIPS*, 2018.

[151] Leonid Kantorovitch. On the translocation of masses. *Management Science*, 5(1):1–4, 1958.

[152] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *arXiv preprint*

arXiv:1908.10543, 2019.

[153] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE TPAMI*, 40(7):1711–1725, 2017.

[154] Yen-Liang Lin, Vlad I Morariu, Winston Hsu, and Larry S Davis. Jointly optimizing 3d model fitting and fine-grained classification. In *ECCV*, 2014.

[155] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[156] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

[157] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[158] Seungryong Kim, Dongbo Min, Stephen Lin, and Kwanghoon Sohn. Dctm: Discrete-continuous transformation matching for semantic flow. In *ICCV*, 2017.