Unsupervised Feature Selection for Heterogeneous Data

BY

XIAOKAI WEI
M.Sc. in Mathematics, University of Illinois at Chicago, 2016
B.Eng. in Computer Science, Beijing University of Posts and Telecommunications, 2010

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2017

Chicago, Illinois

Defense Committee:

Dr. Philip S. Yu, Chair and Advisor, Professor at Dept. of Computer Science
Dr. Piotr Gmytrasiewicz, Associate Professor at Dept. of Computer Science
Dr. Yuheng Hu, Assistant Professor at Dept. of Information and Decision Sciences
Dr. Xinhua Zhang, Assistant Professor at Dept. of Computer Science
Dr. Brian Ziebart, Assistant Professor at Dept. of Computer Science

This thesis is dedicated to my dear parents.

ACKNOWLEDGMENTS

First, I would like to thank my Ph.D. advisor, Professor Philip S. Yu, who has provided me guidance and support since 2011. I deeply appreciate his mentoring during my Ph.D. study and I have learned valuable research skills from him. Without his patient advising, I would not have the opportunity to grow as a researcher in data mining in the past few years, and this dissertation would not have been finished. His suggestions and guidance are not only valuable to my academic achievements, but also will help my professional career in future.

Also, I want to thank Professor Piotr Gmytrasiewicz, Professor Yuheng Hu, Professor Xinhua Zhang and Professor Brian Ziebart for their helpful feedback and suggestions on this dissertation. I sincerely appreciate their valuable time to be on my dissertation committee.

Besides, I would like to thank my colleagues and my friends at the University of Illinois at Chicago. I had a lot of fun time with these friends and have learned a lot through inspiring discussions with them. I truly enjoy the collaborations with these highly brilliant researchers.

At last, I want to thank my parents and my grandparents, for their unconditional love and support.

XW

CONTRIBUTION OF AUTHORS

Chapter 2 presents a published manuscript (101), for which I was the primary author. My advisor Professor Philip S. Yu contributed to revising the manuscript. Chapter 3 presents a published manuscript (99), for which I was the primary author. Sihong Xie and Professor Philip S. Yu contributed to revising the manuscripts and discussion with respect to the work. Chapter 4 presents an published manuscript (96), for which I was the primary author, Bokai Cao and Professor Philip S. Yu contributed to revising the manuscript and discussion with respect to the work. Chapter 5 presents an accepcted manuscript (100), for which I was the primary author. Linchuan Xu, Bokai Cao and Professor Philip S. Yu contributed to revising the manuscript and discussion with respect to the work. Chapter 6 presents an accepted manuscript (98), for which I was the primary author. Bokai Cao and Professor Philip S. Yu contributed to revising the manuscript and discussion with respect to the work. Chapter 7 presents a unpublished manuscript (97), for which I was the primary author. Bokai Cao and Professor Philip S. Yu contributed to revising the manuscript and discussion with respect to the work.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF FIGURES

SUMMARY

Nowadays, one is often confronted with the problem of high-dimensional data in many machine learning and data mining applications. Hence, feature selection has become an important technique since it can alleviate curse of dimensionality, speed up learning process and provide better interpretability. In this thesis, I will introduce my research on unsupervised feature selection, with special focus on heterogeneous data, include networked data, multi-view/multi-modal data and data with complex side information.

In the first part of the dissertation, we focus on unsupervised feature selection without relying on clustering. We select the features that maximally preserve stochastic neighbors and this approach achieves state-of-the-art performance.

Also, various social/information networks, such as bibliographic network and gene network become prevalent in the big data era, I introduce three methods designed for feature selection on social/information networks. The first and second approaches study feature selection as a standalone task, while the third approach studies coupled feature selection in the task of representation learning. The first approach proposes partial order preserving principle, which exploits linkage information for feature selection in an efficient manner. The second method considers a generative view for feature selection on networks, in which we assume the node features and links are generated from a set of high quality features. The third model studies joint representation learning and feature selection with cross-view link prediction on partially observable networks as an application.

SUMMARY (Continued)

As multi-view/multi-modal data become ubiquitous in the era of big data, we study how to perform effective unsupervised feature selection for multi-view data. We first fuse the information from multi-view data by cross diffusion. Then matrix alignment is performed to use the fused information for feature selection on each view.

Moreover, data are often equipped with complex side information, in which different objects are inter-connected and can be represented by heterogeneous information network. We study how to extract information from such side information and how to utilize it to guide feature selection.

CHAPTER 1

INTRODUCTION

1.1 Dissertation Framework

In the era of big data, one is often confronted with the problem of high-dimensional data in many machine learning and data mining applications. Hence, feature selection has become an important technique since it can alleviate curse of dimensionality, speed up learning process and provide better interpretability. This dissertation work focuses on unsupervised feature selection as class labels are usually expensive to obtain.

In unsupervised feature selection, it is typically more challenging to evaluate the quality of features than its supervised counterpart due to the lack of guidance from class labels. We designed several new criteria, which have some desirable properties and can effectively identify discriminative features without using class labels. Moreover, due to better capability of data collection, data samples usually come in heterogeneous forms, such as networked data, multi-modal/multi-view data and data equipped with complex side information. Such heterogeneous information (e.g., network structure and additional views) can be highly useful when class labels are not available. Meanwhile, how to better utilize the abundant information contained in heterogeneous data poses additional challenges.

Through studying and researching on feature selection for heterogeneous data, we provide insightful analysis and innovative methods to solve existing and new research problems. The dissertation covers the following different research directions related to feature selection on heterogeneous forms of data:

- In order to select discriminative features for traditional data, we design a novel criterion by preserving stochastic neighbors. It does not rely on potentially noisy cluster labels and is able to select high-quality features in unsupervised scenario.
- We propose three approaches for feature selection on network data. To better utilize network structure, we designed two methods: Partial Order Preserving (POP) approach and Generative Feature Selection (GFS) approach. The former method is able to select features in an efficient manner and the latter exploits information both links and node attributes. In addition, we develop a method with joint representation learning and feature selection, to tackle a novel problem of cross view link prediction.
- As multi-view/multi-modal information becomes prevalent, we design a new method, Cross Diffused Matrix Alignment based Feature Selection (CDMA-FS), to better utilize the information from different views. The proposed method effectively fuses multi-view information by cross diffusion. Then, the feature utility can be evaluated in a non-linear manner by performing matrix alignment.
- To leverage more complex form of side information, we introduce a novel method which models the inter-connected side information as heterogeneous information network. The proposed model is able to exploit information from the complex side information through meta-path extraction. Joint graph embedding and feature selection are performed to select high-quality features.

1.2 Unsupervised Feature Selection by Preserving Stochastic Neighbors

(Part of the section was previously published in (101).)

Unsupervised feature selection is more challenging than its supervised counterpart due to the lack of labels. In chapter 2, we present an effective method, Stochastic Neighbor-preserving Feature Selection (SNFS), for selecting discriminative features for traditional data in unsupervised setting. We employ the concept of stochastic neighbors and select the features that can best preserve such stochastic neighbors by minimizing the Kullback-Leibler (KL) Divergence between neighborhood distributions. The proposed approach measures feature utility jointly in a non-linear way and discriminative features can be selected due to its 'push-pull' property. We develop an efficient algorithm for optimizing the objective function based on projected quasi-Newton method. Moreover, few existing methods provide ways for determining the optimal number of selected features and this hampers their utility in practice. Our approach is equipped with a guideline for choosing the number of features, which provides nearly optimal performance in our experiments. Experimental results show that the proposed method outperforms state-of-the-art methods significantly on several real-world datasets.

1.3 Efficient Partial Order Preserving Unsupervised Feature Selection on Networks

(Part of the section was previously published in (99).)

In the past decade, research on network data has attracted much attention and many interesting phenomena have been discovered. Such data are often characterized by high dimensionality but how to select meaningful and more succinct features for network data received relatively less attention. In chapter 3, we investigate unsupervised feature selection problem on networks. To effectively incorporate linkage information, we propose a Partial Order Preserving (POP) principle for evaluating features. We show the advantage of this novel formulation in several respects: effectiveness, efficiency and its connection to optimizing AUC. We propose three instantiations derived from the POP principle and

evaluate them using three real-world datasets. Experimental results show that our approach has significantly better performance than state-of-the-art methods under several different metrics.

1.4 Unsupervised Feature Selection on Networks: A Generative View

(Part of the section was previously published in (96).)

Most existing feature selection methods fail to incorporate the linkage information, and the state-of-the-art approaches usually rely on pseudo labels generated from clustering. Such cluster labels may be far from accurate and can mislead the feature selection process. In chapter 4, we investigate the problem of unsupervised feature selection on networks from a generative point of view. We propose a generative point of view for unsupervised features selection on networks that can seamlessly exploit the linkage and content information in a more effective manner. We assume that the link structures and node content are generated from a succinct set of high-quality features, and we find these features through maximizing the likelihood of the generation process. Experimental results on three real-world datasets show that our approach can select more discriminative features than state-of-the-art methods.

1.5 Learning Representation Consensus with Coupled Feature Selection for Cross View Link Prediction

(Part of the section was previously published in (100).)

Link Prediction has been an important task for studying information and social networks and most existing approaches assume the completeness of network structure. However, in many real-world networks, the links and node attributes can usually be partially observable. In chapter 5, we study the problem of **Cross View Link Prediction (CVLP)** on partially observable networks, where the focus is to recommend nodes with only links to nodes with only attributes (or vice versa). We aim to bridge

the information gap by learning a robust consensus for link-based and attribute-based representations so that nodes become comparable in the latent space. Moreover, feature selection is performed jointly with the representation learning to alleviate the effect of noisy high-dimensional attributes. We present two instantiations of this framework with different loss functions and develop an alternating optimization framework to solve the problem. Experimental results on four real-world datasets show the proposed algorithm outperforms the baseline methods significantly for cross-view link prediction.

1.6 Multi-view Unsupervised Feature Selection by Cross-diffused Matrix Alignment

(Part of the section was accepted and to appear as (98).)

Multi-view high-dimensional data become increasingly popular in the big data era. Feature selection is a useful technique for alleviating the curse of dimensionality. In chapter 6, we study unsupervised feature selection for multi-view data. Traditional feature selection methods are mostly designed for single-view data and cannot fully exploit the rich information from multi-view data. Existing multi-view feature selection methods are usually based on noisy cluster labels which might not preserve sufficient information from multi-view data. To better utilize multi-view information, we propose a method, CDMA-FS, to select features for each view by performing alignment on a cross diffused matrix. Experiments results on four real-world datasets show that the proposed method is more effective than the state-of-the-art methods in multi-view setting.

1.7 Unsupervised Feature Selection with Complex Side Information

Many datasets are also equipped with certain side information of complex structure. Such side information can be critical for feature selection when class labels are unavailable. In chapter 7, we propose a new feature selection method, SideFS, to exploit such rich side information. We model

the complex side information as a heterogeneous network and derive instance correlations to guide subsequent feature selection. Representations are learned from the side information network and the feature selection is performed in a unified framework. An alternating method is developed for SideFS to solve the optimization problem. Experimental results show that the proposed method can effectively enhance the quality of selected features by incorporating complex side information.

CHAPTER 2

UNSUPERVISED FEATURE SELECTION BY PRESERVING STOCHASTIC NEIGHBORS

(This chapter was previously published as "Unsupervised Feature Selection by Preserving Stochastic Neighbors", in Proceedings of the 19th international Conference on Artificial Intelligence and Statistics (AISTATS 16), 2016, with permission to reuse.)

2.1 Introduction

In the era of big data, datasets are often characterized by high dimensionality in many machine learning or data mining tasks. To alleviate the curse of dimensionality, feature selection (34) (63) (99) has become an important technique. By selecting a subset of high-quality features, feature selection can speed up the learning process and provide easier interpretation of the problem.

Depending on the availability of supervision information, feature selection methods can be categorized into two classes: supervised feature selection and unsupervised feature selection. Since class labels are usually expensive to obtain, our work focuses on unsupervised scenario. It is usually more difficult to evaluate the discriminativeness of features without guidance from class labels. Different heuristics (e.g., frequency based, variance based) have been proposed to perform unsupervised feature selection. Similarity-preserving approaches (34) (110) have gained much popularity among others. In such similarity preserving methods, a feature is considered to be good if it can preserve the local manifold structure well.

Recently, pseudo label based algorithms with $L_{2,1}$ norm (106) (48) have become increasingly popular. Since class labels are not available, such methods attempt to generate cluster labels (i.e., pseudo labels) or subspace representations through linear transformation/regression regularized by $L_{2,1}$ norm. They rank features by their usefulness on predicting pseudo label/constructing the subspace. One major drawback of such approach is that the cluster labels are usually far from accurate and such inaccurate pseudo labels can mislead feature selection.

The central issue in unsupervised feature selection is how to effectively uncover the discriminative information embedded in the data. Inspired by the popular visualization technique Stochastic Neighbor Embedding (SNE) (35), we employ the concept of stochastic neighbors for the purpose of unsupervised feature selection. We develop a novel unsupervised feature selection method, Stochastic Neighbor-preserving Feature Selection (SNFS), to select a set of high-quality features. Specifically, for each data point, other data points are its neighbors with certain probability. The goal is to select a set of features that best preserve such stochastic probability. With this criterion, the derived gradient update formula is very simple, and it has a desirable *pull-push* property that the selected features can pull similar data points close and push dissimilar data points far apart. As a result, data points from different classes could be better separated with the set of selected features. The advantages of SNFS can be summarized as follows:

- The aim of unsupervised feature selection is usually to improve subsequent clustering tasks. Popular clustering methods such as KMeans and Spectral Clustering (62) are distance/similarity-based methods: KMeans needs to measure the similarity/distance to centroids when assigning data points and Spectral Clustering needs to build a similarity graph for clustering. State-of-

the-art $L_{2,1}$ norm based approaches (106) (48) (68) select features based on how well they can linearly explain the variance of cluster labels (i.e., by their linear regression coefficients). By contrast, SNFS is not based on linear regression and is able to evaluate features jointly in a more similarity-friendly manner.

- The proposed criterion aims to keep similar data points closer than dissimilar data points. Such a criterion can select discriminative features to make the clusters more separable.
- For supervised feature selection, one can choose the number of selected features based on cross-validation performance. But it is very challenging to choose the optimal number of features in unsupervised setting. The inability of existing approaches (48) (68) to choose optimal feature size limits their practical utility. We provide a guideline for deciding feature sizes and experimental results indicate that this proposed guideline can usually achieve decent performance.

We develop an efficient optimization algorithm for the proposed method based on projected quasi-Newton method. Experimental results on six real-world datasets illustrate the superiority of SNFS.

2.2 Related Work

In this section, we review related work on feature selection.

2.2.1 Supervised Feature Selection

The goal of feature selection is to alleviate the curse of dimensionality, enabling machine learning models to achieve comparable, if not better, performance. Traditional feature selection methods generally fall into three categories: filter model (109) (64), wrapper model (23) and embedded model (17) (87). In supervised feature selection, the criterion for feature quality is usually straightforward:

high-quality features should be highly correlated with class labels. Different methods are proposed to capture the correlation between label and feature, such as Mutual Information, Fisher Score (22) and HSIC (78). For example, Song et al. (2007) introduces Hilbert-Schmidt Independence Criterion (HSIC) as a measure of dependence between the features and the labels (78). LASSO (87), as an embedded model, performs feature selection during regression/classification by using L_1 regularization.

2.2.2 Unsupervised Feature Selection

In the unsupervised setting, various heuristics are proposed to guide the feature selection process. One popular guiding principle is to preserve the local manifold structure or similarity (34) (109) (110). But features useful for preserving similarity are not necessarily discriminative. Also, these earlier unsupervised feature selection algorithms tend to evaluate the importance of features individually (34) (109), which neglects correlation among features and may introduce redundancy in the selected features. Recent methods (106) (48) (76) (96) using sparsity-inducing norms overcome this issue by evaluating the features as a whole. For example, Unsupervised Discriminative Feature Selection (UDFS) (106) introduces pseudo label-based regression to better capture discriminative information. Sparsity-inducing $L_{2,1}$ norm is used to select the feature jointly. Robust Unsupervised Feature Selection (RUFS) (68) further employs robust $L_{2,1}$ loss on the regression objective to alleviate the effect of outlier instances. Robust Spectral Feature Selection (RSFS) (76) uses robust learning framework with local kernel regression for generating pseudo-labels.

Essentially, all the pseudo label based methods evaluate the utility of features based on how well in linear projection they can explain the variance of the cluster labels. As a result, they have similar drawbacks: first, they only evaluate features on their linear ability and overlook their non-linear usefulness.

Second, the pseudo labels derived from clustering are usually not accurate enough. The noisy information contained in the pseudo labels can further mislead feature selection. Moreover, state-of-the-art pseudo-label approaches (68) (76) usually have $3 \sim 5$ free parameters (e.g., neighborhood size, number of latent dimensions and hyperparameters controlling regularization terms), which are difficult, if not impossible to tune without supervision.

2.3 Formulations

2.3.1 Notations

Suppose we have n data samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and the total number of features is D. So $\mathbf{x}_i \in \mathbb{R}^D$ and x_{it} denotes the value of t-th ($t = 1, \ldots, D$) feature of $\mathbf{x}_i$. Our goal is to select d ($d \ll D$) discriminative features. We use $\mathbf{w} \in \{0, 1\}^D$ as the selection indicator vector: $w_t = 1$ indicates the t-th feature is selected and $w_t = 0$ otherwise.

2.3.2 Stochastic Neighbors-preserving Feature Selection

We assume each data sample has all the other data samples as stochastic neighbors with certain probability, rather than having a fixed set of neighbors. Let us denote the probability of $\mathbf{x}_i$ having $\mathbf{x}_j$ ($j \neq i$) as its neighbors as p_{ij} and assume p_{ij} depends on their similarity S_{ij}. The larger S_{ij} is, the more likely $\mathbf{x}_j$ is $\mathbf{x}_i$'s neighbor. For convenience, we also define $p_{ii} = 0$ for $i = 1, \ldots, n$.

To make $\sum_{j=1}^{n} p_{ij} = 1$, we use the softmax function to define this probability.

$$p_{ij} = \frac{\exp(S_{ij})}{\sum_{k \neq i} \exp(S_{ik})} \tag{2.1}$$

In principle, S_{ij} could be any affinity measure, such as cosine similarity and negative euclidean distance. We use inner product to measure the similarity of two data points and therefore $S_{ij} = \mathbf{x}_i^T \mathbf{x}_j$.

To add more flexibility to the model, we also include a scale (bandwidth) parameter σ^2 in the soft-max function as follows. We will discuss how to set this parameter later in this chapter.

$$p_{ij} = \frac{\exp\left(S_{ij}/\sigma^2\right)}{\sum_{k \neq i} \exp\left(S_{ik}/\sigma^2\right)} \tag{2.2}$$

After feature selection, we denote similarity calculated on the selected features as $s_{ij} = \mathbf{x}_i^T \text{diag}(\mathbf{w}) \mathbf{x}_j$, where $\text{diag}(\mathbf{w})$ is the diagonal matrix using $\mathbf{w}$ as diagonal elements. So, the probability of $\mathbf{x}_j$ being the neighbor of $\mathbf{x}_i$ after feature selection is q_{ij}.

$$q_{ij} = \frac{\exp\left(\frac{\mathbf{x}_i^T \text{diag}(\mathbf{w}) \mathbf{x}_j}{\sigma^2}\right)}{\sum_{k \neq i} \exp\left(\frac{\mathbf{x}_i^T \text{diag}(\mathbf{w}) \mathbf{x}_k}{\sigma^2}\right)} \tag{2.3}$$

Note that q_{ij} (or p_{ij}) is not only influenced by s_{ij} (or S_{ij}), but also affected by s_{ik} (or S_{ij}, $k = 1, \ldots, j-1, j+1, \ldots, n$) via the normalization term. Therefore, q_{ij} (or p_{ij}) is determined by the relative value of s_{ij} (or S_{ij}) compared with other s_{ik} (or S_{ik}).

To preserve the stochastic neighbors, we try to make two distributions $\mathbf{q}_i = [q_{i1}, \ldots, q_{in}]^T$ and $\mathbf{p}_i = [p_{i1}, \ldots, p_{in}]^T$ similar by minimizing their KL divergence for each $\mathbf{x}_i$.

$$KL(\mathbf{p}_i || \mathbf{q}_i) = \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{2.4}$$

We propose the following feature selection criterion: selecting the set of features to minimize the sum of KL divergence between $\mathbf{p}_i$ and $\mathbf{q}_i$ on all the data points.

$$\begin{aligned} &\min_{\mathbf{w}} \sum_{i=1}^{n} \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}} \\ &s.t. \sum_{t=1}^{D} w_t = d \\ &w_t \in \{0,1\}, \forall t = 1, \ldots, D \end{aligned} \tag{2.5}$$

The goal is that, for similar data points, we still want them to be similar after feature selection. For dissimilar data points, it is desirable to keep them dissimilar with selected features. So, by minimizing KL-divergence between $\mathbf{p}_i$ and $\mathbf{q}_i$ for $i = 1, \ldots, n$, we select the features that make similar data samples still closer than dissimilar samples.

2.3.3 Setting Scale Parameter

In this subsection, we discuss how to set the scale/bandwidth parameter σ^2. $\mathbf{p}_i$ is influenced by the value of σ^2: the higher σ^2 is, the higher entropy $\mathbf{p}_i$ has. For data sample $\mathbf{x}_i$, when σ^2 is relatively large, other data samples tend to have similar probability of being $\mathbf{x}_i$'s neighbors. In the extreme case, when σ^2 goes to infinity, all other data samples have equal probability of being $\mathbf{x}_i$'s neighbor. When σ^2 is small, the probability tends to be concentrated on a small number of most similar neighbors. We define the average perplexity as follows.

$$Perplexity(P) = 2^{\frac{1}{n} \sum_{i=1}^{n} H(\mathbf{p}_i)} \tag{2.6}$$

where $H(\mathbf{p}_i) = -\sum_{j \neq i} p_{ij} \log p_{ij}$ is the entropy of $\mathbf{p}_i$. The perplexity has a more intuitive interpretation than σ^2: it can be interpreted as a smooth measure of the effective number of neighbors. The perplexity is a monotonically increasing function of σ^2 and larger perplexity corresponds to larger σ^2. After we specify the value of perplexity, the value of σ^2 can be found by line search (e.g., binary search). So we do not need to directly set σ^2. Rather, we use perplexity as a proxy since it has more intuitive explanation. As we will show in the experimental results, SNFS can usually achieve good performance for a reasonably large range of perplexity (e.g., $5 \sim 50$).

2.4 Optimization

2.4.1 Gradient Derivation

The formulation in Equation 2.5 is a '0/1' integer programming problem, which is time-consuming to optimize. To make the optimization more efficient, we relax the '0/1' constraint on w_t ($\forall t = 1, \ldots, D$) to real values in the range of $[0, 1]$. Also, we re-write the summation constraint $\sum_{t=1}^{D} w_t = d$ using Lagrangian multiplier.

$$\begin{aligned} &\min_{\mathbf{w}} \sum_{i=1}^{n} \sum_{j \neq i} p_{ij} \log \frac{p_{ij}}{q_{ij}} + \lambda ||\mathbf{w}||_1 \\ &s.t.\ 0 \leq w_t \leq 1, \forall t = 1, \ldots, D \end{aligned} \tag{2.7}$$

where $|| \cdot ||_1$ is the L_1 norm and λ is the parameter to control the L_1 regularization. Note that in general L_1 norm is not differentiable due to the non-smoothness at value 0, but in our case, $|w_t| = w_t$ since w_t ($\forall t = 1, \ldots, D$) is always non-negative.

Let us denote the objective in Equation 2.7 as $\mathcal{L}$. It takes several steps to calculate the gradient $\frac{\partial \mathcal{L}}{\partial w_t}$, but the final result is simple.

$$\frac{\partial \mathcal{L}}{\partial w_t} = -\sum_{i=1}^{n}\sum_{j \neq i}(p_{ij} - q_{ij})x_{it}x_{jt}/\sigma^2 + \lambda \tag{2.8}$$

If we use negative euclidean distance as the affinity measure (i.e., $s_{ij} = -(\mathbf{x}_i - \mathbf{x}_j)^T \text{diag}(\mathbf{w})(\mathbf{x}_i - \mathbf{x}_j)$), one can derive the following gradient formula in a similar manner:

$$\frac{\partial \mathcal{L}}{\partial w_t} = \sum_{i=1}^{n}\sum_{j \neq i}(p_{ij} - q_{ij})(x_{it} - x_{jt})^2/\sigma^2 + \lambda \tag{2.9}$$

Such a gradient update formula in Equation 2.8 (or Equation 2.9) has an intuitive *push-pull* interpretation: when $\mathbf{x}_j$ is more likely to be $\mathbf{x}_i$'s neighbor than desired (e.i., $p_{ij} < q_{ij}$), w_t is updated in the direction of $x_{it}x_{jt}/\sigma^2$ (or $-(x_{it} - x_{jt})^2/\sigma^2$) to push them away; when $\mathbf{x}_j$ is less likely to be $\mathbf{x}_i$'s neighbor than desired (e.i., $p_{ij} > q_{ij}$), w_t is updated to pull them closer. If a feature has little contribution in preserving the distribution of stochastic neighbors, its weight tends to shrink to zero under the effect of L_1 regularization.

2.4.2 Projected Quasi-Newton Method

To make the optimization more efficient, we incorporate second order information by using projected quasi-Newton method (6). At each iteration, we partition w_t ($t = 1, \ldots, D$) into two groups: restricted variables $\mathcal{R}_w$ and free variables $\mathcal{F}_w$.

$$\begin{aligned}\mathcal{R}_w =&\{w_t | (w_t \leq \epsilon \wedge \frac{\partial \mathcal{L}}{\partial w_t} > 0) \text{ or} \\ &(w_t \geq 1 - \epsilon \wedge \frac{\partial \mathcal{L}}{\partial w_t} < 0)\}\end{aligned} \tag{2.10}$$

$$\mathcal{F}_w = \{w_1, w_2, \ldots, w_D\} - \mathcal{R}_w \tag{2.11}$$

where ϵ is a small positive value. The restricted variables are those close to the lower or upper bound in their gradient direction. In Newton's Method, the scaling matrix $\bar{\mathbf{S}}^k$ for the free variables at iteration k is the inverse Hessian matrix.

$$\bar{\mathbf{S}}^k = [\nabla^2 \mathcal{L}(\mathbf{w}^k)]^{-1}_{\mathcal{F}^k_w}, \tag{2.12}$$

For both free and restricted variables, the scaling matrix can be defined as follows.

$$\mathbf{S}^k = \begin{bmatrix} \bar{\mathbf{S}}^k & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \tag{2.13}$$

The scaling matrix $\mathbf{D}$ for restricted variables can be identity matrix. In each iteration, we find appropriate step size η^k using backtracking line search to satisfy Armijo rule:

$$f(\mathbf{w}^k) - f(\mathbf{w}^k + \eta^k \mathbf{d}^k) \leq c_1 \eta^k \mathbf{d}^k \tag{2.14}$$

where c_1 is a constant in the range of $0 \leq c_1 \leq 1$ and $\mathbf{d}$ is the descent direction ($\mathbf{d} = \mathbf{S}^k \nabla \mathcal{L}(\mathbf{w}^k)$ in our case). Note that computing the step size does not increase the computational complexity of the method, since computing the orthogonal projection after each backtracking step is trivial.

The final projected-Newton update formula is as follows.

$$\mathbf{w}^{k+1} \leftarrow \mathcal{P}(\mathbf{w}^k - \eta^k \mathbf{S}^k \nabla \mathcal{L}(\mathbf{w}^k)) \tag{2.15}$$

where the projection operator $\mathcal{P}(\cdot)$ projects the value $(\cdot)$ to $[0, 1]$.

$$[\mathcal{P}(\mathbf{w})]_t = \min(1, \max(0, w_t)), \forall t = 1, \ldots, D \tag{2.16}$$

For restricted variables $w_t^k \in \mathcal{R}_w^k$, we can directly set them to 0 or 1 if ϵ is sufficiently small.

$$[\mathcal{P}(\mathbf{w}^k - \eta^k \mathbf{S}^k \nabla \mathcal{L}(\mathbf{w}^k))]_t = \begin{cases} 0, & \text{if } w_t \leq \epsilon \wedge \frac{\partial \mathcal{L}}{\partial w_t} > 0 \\ 1, & \text{if } w_t \geq 1 - \epsilon \wedge \frac{\partial \mathcal{L}}{\partial w_t} < 0 \end{cases} \tag{2.17}$$

So, we only need to compute the scaling matrix $\bar{\mathbf{S}}$ for free variables. This can save considerable computation time if the number of free variables is small (i.e., $|\mathcal{F}_w| \ll |\mathcal{R}_w|$), which is usually the case in feature selection scenario. It has been shown (6) (26) this projected-Newton method is convergent under mild conditions.

Theorem 2.4.1 *For a loss function $\mathcal{L}$, assume that $\nabla \mathcal{L}$ is Lipschitz continuous and $\nabla^2 \mathcal{L}$ has bounded eigenvalues. Then every limit point of $\mathbf{w}^k$ generated by Equation 2.15 is a stationary point of Equation 2.7.*

However, the Newton-step is often computation-intensive and requires D^2 storage. To save computation time and storage space, we approximate the Hessian with L-BFGS, as shown in Algorithm 1. L-BFGS only requires $O(mD)$ storage if the gradients in that last m iterations are used. Though the convergence rate of the method has been shown for $\mathbf{S}^k$ derived from the Hessian, the convergence itself only requires a positive-definite gradient scaling $\mathbf{S}^k$ with bounded eigenvalues for all k (6). Thus, quasi-Newton approximations (e.g., L-BFGS) can also be employed to derive convergent methods. In our experiments, the optimization algorithm usually converges in less than 20 iterations.

Algorithm 1 Projected L-BFGS Algorithm for SNFS

1: Initialize $\mathbf{w} \leftarrow [1, 1, \ldots, 1]$
2: **while** not converge **do**
3: Identify restricted and free variables by Equation 2.10 and Equation 2.11.
4: Set the restricted variables to the corresponding lower or upper bound (i.e., 0 or 1)
5: Calculate the gradient using **??** and $\bar{\mathbf{S}}$ for free variables using the gradient information of last m iterations.
6: Use backtracking line search to find the step size η that satisfies Armijo condition Equation 2.14
7: Update $\mathbf{w}$ using formula Equation 2.15
8: **end while**
9: Select features with w_t ($t = 1, \ldots, D$) greater than $1 - \alpha$

2.4.3 Determining the number of selected features

It is worth noting that $\mathbf{w}$ can be intuitively interpreted as features' importance scores in preserving stochastic neighbors. The relaxed w_t ($t = 1, \ldots, D$) has the maximum value of 1 and the minimum of 0. For unrelaxed $\mathbf{w}$, we simply retain the features with $w_t = 1$. Similarly, to select the high-quality

TABLE I: Comparison of different similarity-based unsupervised feature selection methods

Methods	LS(34), SPEC(109)	SPFS(111), MCFS(12)	NDFS(48), RUFS(68), RSFS(76)	SNFS
Evaluate features jointly?	×	✓	✓	✓
Evaluate features non-linearly?	✓	×	×	✓
Do not rely on clustering?	✓	✓	×	✓
Guideline for setting number of selected features?	×	×	×	✓

features from relaxed version of $\mathbf{w}$, we can select the features with w_t equal or close to 1. For example, we can keep the features with w_t greater than $(1-\alpha)$ for a small α (e.g., $\alpha = 0.05$ or $\alpha = 0.1$. We denote the number of features with scores larger than $(1-\alpha)$ as $N_{1-\alpha}$. For example, $N_{0.9}$ is the number of features that have scores greater than 0.9. As we will show in the experimental results, such a strategy can usually achieve near-optimal performance.

Since $N_{0.9}$ is influenced by the regularization parameter λ (i.e., larger λ leads to smaller $N_{0.9}$), one can also do a line search for appropriate λ (e.g., via binary search) if he wants to retain a specific number of features.

2.5 Discussion

Similarity-based approaches are a popular thread of unsupervised feature selection methods. In this section, we discuss how SNFS is different and superior to other similarity-based methods.

Laplacian Score (34) and SPEC (109) are based on the eigenvalues of similarity matrix. They assign a score to each feature and select the features with higher scores. Features are evaluated individually and redundancy can have negative impact on the performance of selected features. Besides, while the

selected features will make similar data points still similar, they make little effort to make dissimilar data points far apart.

SPFS and MCFS (12) perform sparse linear regression towards the spectral decomposition of similarity matrix and choose the features with large coefficients. NDFS (48), RUFS (68) and RSFS (76) generate cluster labels and perform linear regression with $L_{2,1}$ norm. The drawback is that the inaccurate cluster labels can provide misleading information for feature selection. In all these regression-based methods, the selection criterion depends on how well the features can linearly explain the variance of cluster labels/subspace representation. This limits their effectiveness in clustering tasks, since most popular clustering algorithms are based on similarity/distance, such as KMeans and Spectral Clustering (62).

Moreover, a common shortcoming of all these methods is that they do not provide any guideline for choosing the number of selected features.

In contrast, SNFS evaluates features jointly and non-linearly. Rather than preserving the similarity itself, SNFS focuses on preserving the relative value of similarity in each neighborhood. Table I summarizes the difference between several popular similarity-based feature selection methods. We can see that SNFS has several desirable properties, which enable it to identify a set of more discriminative features.

TABLE II: Statistics of datasets

Statistics	BBC	BBC Sport	BlogCatalog	TDT	Guardian	Newsgroup
# of instances	2225	737	500	1500	302	1575
# of features	9636	4612	4547	6458	3631	2849
# of classes	5	5	5	15	6	4

2.6 Experiment

2.6.1 Baselines

We compare our approach to using all features and five unsupervised feature selection methods as baselines. LS and MCFS are manifold-preserving/similarity-preserving approaches. UDFS, RUFS and RSFS are pseudo-label based methods which also consider the similarity information.

- All Features: It uses all the features for evaluation.
- Laplacian Score (LS): Laplacian score (34) selects the features which can best preserve the local manifold structure.
- MCFS: Multi-cluster Feature Selection (12) performs spectral analysis and sparse regression to select features.
- UDFS: Unsupervised Discriminative Feature Selection (106) is a psuedo-label based approach which performs $L_{2,1}$-norm regularized subspace learning.
- RUFS: Robust Unsupervised Feature Selection (68) generates psuedo labels by NMF (Non-negative Matrix Factorization) and local learning-based regularization (31).

- RSFS: Robust Spectral Feature Selection (76) selects features by robust spectral analysis framework and $L_{2,1}$-norm regularized regression.

2.6.2 Datasets

We use six publicly available datasets: BBC and BBCSport news dataset[1], Guardian news dataset[2], BlogCatalog[3] blog-posts dataset, Newsgroup [4] and TDT2[5]. The statistics of six datasets are summarized in Table II.

2.6.3 Experimental Setting

In this section, we evaluate the quality of selected features by their clustering performance. We use Accuracy and Normalized Mutual Information (NMI) to evaluate the result of clustering, following the typical setting of evaluation for unsupervised feature selection (106) (48). These two metrics evaluate the cluster quality by matching and comparing the cluster labels with ground-truth labels (more detailed definition of the two metrics is presented in supplemental material). Higher values of Accuracy and NMI indicate better quality of clustering.

We set $k = 5$ for the kNN neighbor size in the baseline methods following previous convention (48). For the number of pseudo classes in UDFS, RUFS and RSFS, we use the ground-truth number of

---

[1] http://mlg.ucd.ie/datasets/bbc.html

[2] http://mlg.ucd.ie/datasets/3sources.html

[3] http://dmml.asu.edu/users/xufei/datasets.html

[4] http://www.cs.umb.edu/~smimarog/textmining/datasets/

[5] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

classes. Besides, UDFS, RUFS and RSFS also require specifying the values of several other regularization parameters. In the original papers of UDFS, RUFS and RSFS, they use class labels to find the best parameters by grid search. However, this violates the assumption of no supervision and could be unfair to approaches with less or no free parameters. Nonetheless, we perform grid search in the range of $\{0.1, 1, 10\}$ for the regularization parameters in UDFS, RUFS and RSFS. Besides the best performance, we also report the median performance for them, which is a more realistic reflection of these methods' practical power. For SNFS, we fix $perplexity = 15$ and $\lambda = 10^{-3} \times n$ on all datasets and we will discuss the sensitivity of these two parameters in the following subsection. [1].

Following the convention in previous work (12) (106), we use KMeans [2] for clustering evaluation. Since Kmeans is affected by the initial seeds, we repeat the experiment for 20 times and report the average performance. We vary the number of features in the range of $\{100, 200, 400, 600\}$. For SNFS, we report the clustering performance using the features with scores greater than 0.9.

2.6.4 Clustering Results

The clustering accuracy on six datasets is shown in Table III. The experimental results show that feature selection is a very effective technique for enhancing clustering. With much less features, SNFS ($N_{0.9}$) can obtain better accuracy and NMI than using all the features. For instance, compared with using all 4547 features, SNFS with only 230 features improves the clustering accuracy by 36.4% on

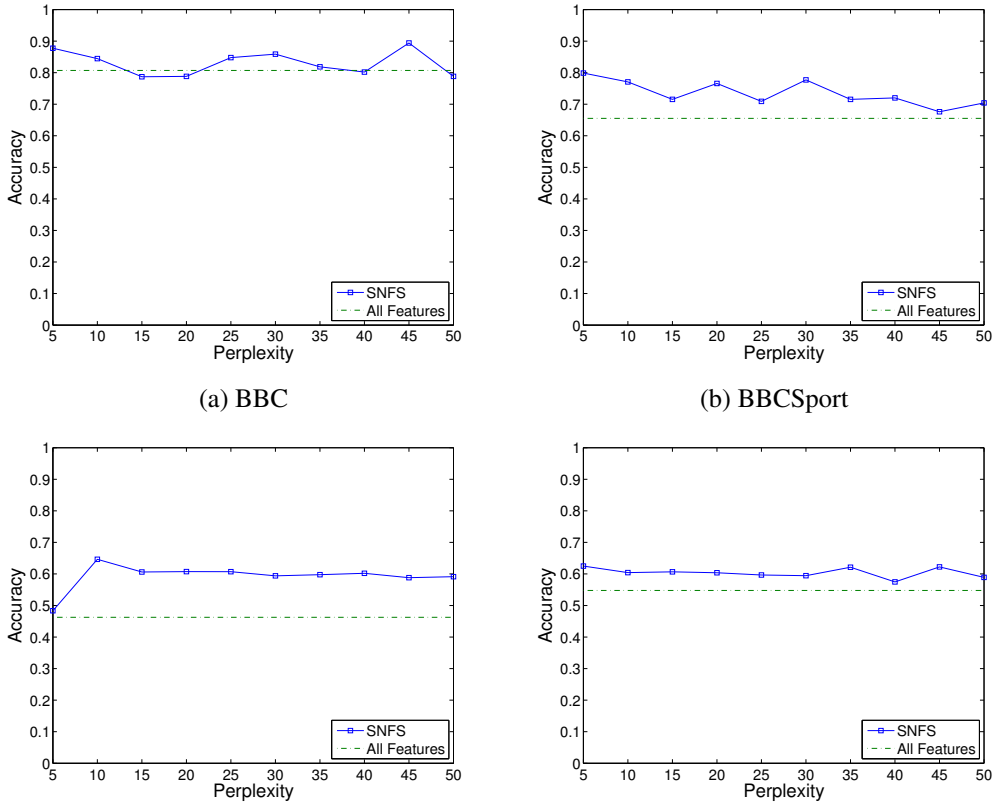
---

[1] For the projected quasi-Newton method in the optimization of SNFS, we use the implementation at `http://www.cs.ubc.ca/~schmidtm/Software/minConf.html`

[2] We use the code at `http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html`

TABLE III: Clustering accuracy on six datasets. For UDFS, RUFS, RSFS, median/best performance is reported. SNFS($N_{0.9}$) denotes the performance of SNFS with top $N_{0.9}$ features.

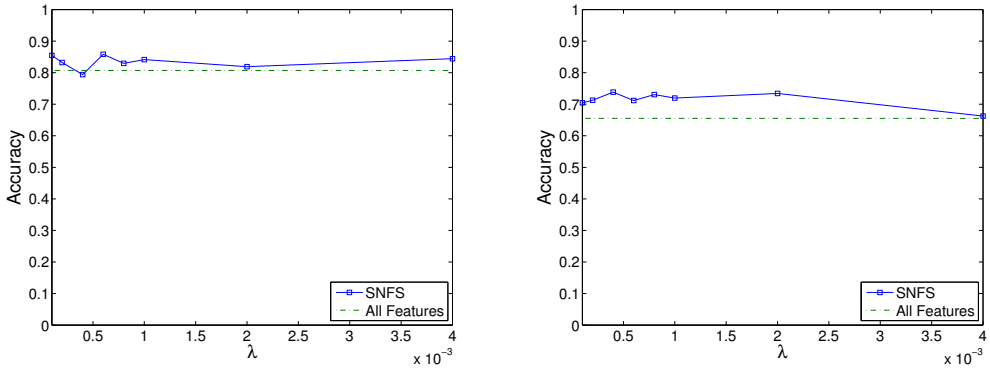
Method	**BBC**				**BBC Sport**			
# features	100	200	400	600	100	200	400	600
All Features	0.8071				0.6551			
LS	0.2360	0.2655	0.4322	0.4718	0.4185	0.4561	0.5110	0.6751
MCFS	0.6223	0.7489	0.7793	0.8217	0.6082	0.7075	0.7027	0.7248
UDFS	0.4246/0.4811	0.6174/0.6681	0.7766/0.7805	0.7599/0.7763	0.4661/0.4875	0.4770/0.5601	0.5390/0.605	0.5770/0.6139
RUFS	0.4744/0.7548	0.6708/0.8584	0.7976/0.8991	0.8263/0.8836	0.6018/0.7487	0.6598/0.7683	0.7009/0.7518	0.6812/0.7187
RSFS	0.5677/0.7660	0.7523/0.8118	0.8068/0.8863	0.8326/0.8693	0.6158/0.6658	0.6546/0.714	0.6648/0.6961	0.6494/0.7030
SNFS	0.6040	0.7729	0.8165	0.8102	0.5847	0.6881	0.7455	0.6964
SNFS($N_{0.9}$)	0.8414(550)				0.7195(440)			
Method	**BlogCatalog**				**Guardian**			
# features	100	200	400	600	100	200	400	600
All Features	0.4627				0.5477			
LS	0.2998	0.4084	0.4203	0.4003	0.3364	0.4083	0.6573	0.6237
MCFS	0.3704	0.4428	0.4143	0.4161	0.5093	0.5053	0.5361	0.5348
UDFS	0.3901/0.3917	0.4069/0.4691	0.4383/0.4749	0.4876/0.5321	0.3682/0.4998	0.4525/0.5144	0.5127/0.5394	0.5247/0.5411
RUFS	0.4877/0.5307	0.5508/0.5756	0.5476/0.5889	0.5375/0.5648	0.4369/0.5608	0.5329/0.5659	0.5490/0.5661	0.5563/0.5791
RSFS	0.3847/0.4969	0.4371/0.5346	0.4709/0.5464	0.5031/0.5412	0.5320/0.5553	0.5296/0.5816	0.5550/0.5907	0.5541/0.5921
SNFS	0.5842	0.6350	0.5924	0.5821	0.5288	0.6063	0.6290	0.6071
SNFS($N_{0.9}$)	0.6313(230)				0.6270(440)			
Method	**Newsgroup**				**TDT**			
# features	100	200	400	600	100	200	400	600
All Features	0.7184				0.7711			
LS	0.2808	0.3863	0.6420	0.7063	0.6548	0.7472	0.7870	0.7816
MCFS	0.3374	0.4368	0.4883	0.5059	0.6128	0.6656	0.7250	0.7367
UDFS	0.3516/0.4145	0.3954/0.4173	0.4403/0.4653	0.4604/0.6335	0.4863/0.4979	0.6102/0.6110	0.7231/0.7247	0.7499/0.7520
RUFS	0.4687/0.6073	0.4595/0.6435	0.4915/0.6476	0.5295/0.6477	0.4381/0.6865	0.5423/0.8112	0.6731/0.7967	0.7614/0.8198
RSFS	0.3969/0.6045	0.4776/0.6516	0.6069/0.6765	0.6225/0.6923	0.6589/0.7806	0.7730/0.8153	0.7695/0.8173	0.7854/0.8261
SNFS	0.4518	0.5075	0.6833	0.7039	0.7502	0.7902	0.7835	0.7890
SNFS($N_{0.9}$)	0.8007(495)				0.8161(163)			



(a) BBC (b) BBCSport

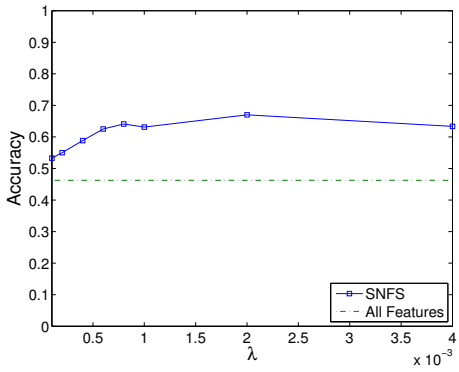
(c) BlogCatalog (d) Guardian

Figure 1: Clustering accuracy with different perplexity values

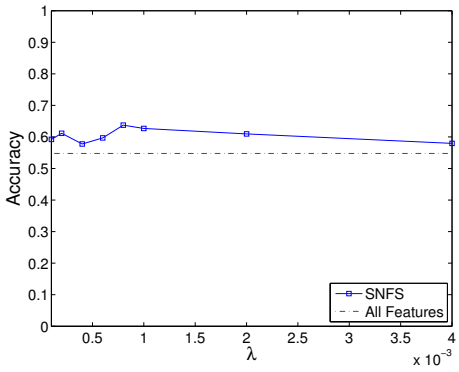


(a) BBC

(b) BBCSport



(c) BlogCatalog



(d) Guardian

Figure 2: Clustering accuracy with different λ

BlogCatalog dataset. Besides the improved accuracy and NMI, using selected features rather than all features can also lead to better interpretability.

We can observe that for SNFS, using $N_{0.9}$ features usually performs the best (or nearly the best) among different number of features. The top $N_{0.9}$ features all have scores equal to 1 or very close to 1. So it is not wise to use only a subset of them. Also, using more than $N_{0.9}$ features may lead to redundancy since less important features are included.

When comparing SNFS with the baseline methods, we observe that SNFS has very competitive performance. The accuracy and NMI of SNFS ($N_{0.9}$) is better than or comparable to the best performance of two strong baselines (RUFS and RSFS) and outperforms their median performance significantly. Since in practice one cannot know the optimal parameters of RUFS and RSFS in unsupervised scenario, the median performance is more representative of their practical utility. Also, all the baseline methods do not provide guidelines for determining the number of selected features. For example, RUFS achieves its top median performance with 400, 200 and 600 features on BBCSport, BlogCatalog and Guardian datasets, respectively. This makes these baseline methods less favorable in practice.

In summary, although these baseline methods also attempt to exploit similarity information in certain ways, they do not perform as well as SNFS. The experimental results illustrate that SNFS is a more effective method for selecting discriminative features.

2.6.5 Sensitivity Analysis

For unsupervised feature selection, it is important that the feature selection algorithm is not very sensitive to its parameters. SNFS has two free parameters: the perplexity and the regularization param-

eter λ for L_1 norm. In this section, we investigate how the performance of SNFS ($N_{0.9}$) varies w.r.t different parameter values.

Figure 1 shows the clustering accuracy of SNFS ($N_{0.9}$) over different values of perplexity. We can observe that SNFS has consistently good performance with different perplexity values ranging from $5 \sim 50$. In most cases, the performance is better than using all the features.

For λ, we vary its value in the range of $[0.0001, 0.0002, 0.0004, 0.0006, 0.0008, 0.001, 0.002, 0.004] \times n$. The clustering performance of SNFS ($N_{0.9}$) over different λ is shown in Figure 2. On most datasets, SNFS has decent performance and can outperform using all features, if λ is not too small ($< 2 \times 10^{-4} \times n$) or too large ($> 2 \times 10^{-3} \times n$).

CHAPTER 3

EFFICIENT PARTIAL ORDER PRESERVING UNSUPERVISED FEATURE SELECTION ON NETWORKS

(This chapter was previously published as "Efficient Partial Order Preserving Unsupervised Feature Selection on Networks", in Proceedings of 2015 SIAM International Conference on Data Mining (SDM 15), with the permission to reuse from Society for Industrial and Applied Mathematics (SIAM))

3.1 Introduction

In many machine learning tasks, one is often confronted with the problem of high dimensionality. Hence, feature selection (34) (63) has become an important technique since it can help alleviate the curse of dimensionality and speed up the learning process. Depending on the availability of class labels, feature selection algorithms can be classified into supervised methods and unsupervised methods. Our work focuses on unsupervised scenario as class labels are usually expensive to obtain. A variety of approaches has been developed for unsupervised feature selection by following different principles. In recent work, similarity-preserving approaches (34) (110) and regression based approaches using pseudo labels (106) (48) have gained much popularity among others.

Network data has become increasingly popular in the past decade, because of the proliferation of various social and information networks. Social media websites such as Facebook, Twitter have millions of users all across the world. Different forms of information networks, e.g, co-author network, citation

network and protein interaction network, also attract considerable attention to analyze (61) (104) (102) (2).

However, traditional feature selection approaches assume that instances are independent and identically distributed (i.i.d). In relational data or information networks, the instances are implicitly or explicitly related, with certain correlation and dependency. For example, in research collaboration networks, the researchers who collaborate with each other tend to share more similar research topics than researchers with no collaboration. But traditional approaches are not able to exploit such rich information contained in the links. LUFS (84) is the first attempt to incorporate network information for unsupervised feature selection, but it uses the structural information at community level via social dimensions (86) and fails to exploit finer-grained link information. Also, LUFS requires several parameters, which are hard to tune in unsupervised setting.

Moreover, the ever increasing size of network data poses additional challenges to feature selection. For instance, Facebook and Linkedin have more than 1.28 billion[1] and 300 million[2] users as of 2014, respectively. However, state-of-the-art unsupervised feature selection methods (106) (48) (84) are prohibitively slow, as their time complexity is usually cubic of the number of features or instances. This makes these algorithms unpractical for large-scale and high-dimensional data.

In this chapter, we present a new perspective to address these challenges regarding both effectiveness and efficiency. We propose a Partial Order Preserving (POP) framework, which allows for parameter-

---

[1]http://en.wikipedia.org/wiki/Facebook

[2]http://en.wikipedia.org/wiki/Linkedin

free mathematical formulation and efficient optimization. Rather than simply preserving the similarity or local manifold structure, POP aims to preserve the partial order of similarity. Network data have abundant partial order information: a node is usually more similar to its neighbors than to the other nodes. By exploiting such difference for feature selection, structural information distinguishing neighbors from non-neighbors is incorporated. As a consequence, more discriminative features can be selected. The main contribution of our work can be summarized in the following:

- We propose a new principle for feature selection on networks: Partial Order Preserving (POP) principle, which selects features that best preserve partial orders. As state-of-the-art approaches are mostly pseudo-label based methods using $L_{2,1}$ norm (106) (48) (84), POP brings a new perspective to the problem of unsupervised feature selection.
- As the linkage relationship in the network is neither complete nor noise free, we present three instantiations of the POP principle, which are robust to noisy/incomplete link information and are parameter free in the objective functions.
- We develop a highly efficient and unified optimization algorithm for these three instantiations. This makes our methods applicable to large-scale datasets.
- We evaluate the proposed algorithms on three real world datasets, and show the advantage of our approach over the baseline methods using different metrics.

3.2 Related Work

In this section, we briefly review related work on feature selection (mainly on unsupervised feature selection).

3.2.1 Unsupervised Feature Selection for Traditional Data

In the unsupervised setting, there are various principles to guide the feature selection process. One popular guiding principle is to preserve the local manifold structure or similarity (34) (109) (110). Recently, pseudo label-based framework (106) (48) gained much popularity. Unsupervised Discriminative Feature Selection (UDFS) (106) introduces pseudo labels to better capture discriminative information and sparsity-inducing $L_{2,1}$ norm is used to select the feature in an iterative manner. Non-negative Discriminative Feature Selection (NDFS) (48) performs non-negative spectral analysis and feature selection simultaneously. But both UDFS and NDFS have computation complexity of $O(D^3T + n^2)$ (D is the number of features, T is the number of iterations, n is number of instances) as eigen-decomposition on $D \times D$ matrix is performed in each iteration. This severely refrains them from being applied to high dimensional data such as text or microarray data. Moreover, they have $3 \sim 4$ parameters to be specified in the objective function. In supervised learning, appropriate parameters can be found through grid search but in unsupervised setting, there is no straightforward way to tune the parameters.

3.2.2 Feature Selection for Network Data

Traditional feature selection techniques assume data instances are independent and identically distributed (i.i.d), which is not the case in network data. In recent years, efforts have been made towards feature selection on relational data. (30) addresses supervised feature selection on network data via adding network-based regularization term to enforce similarity between neighbors. (83) explores supervised feature selection on social media data and integrates different types of relations into the feature selection framework. (85) studies co-selection of features and instances in social media since both features and instances can be noisy and irrelevant. (82) investigates unsupervised multi-view feature

selection on social media but it does not utilize link information. Linked Unsupervised Feature Selection (LUFS) (84) is the only unsupervised feature selection method that utilizes link information. LUFS exploits network information through incorporating social dimension based regularization (86) into the UDFS framework (106). So it shares the same downside of UDFS such as too many parameters and high computational cost. Also, in LUFS, network information is utilized at community/cluster level and finer-grained information in the links is ignored. In this chapter, we propose a parameter-free framework for unsupervised feature selection on network data, which is more effective with lower computation burden.

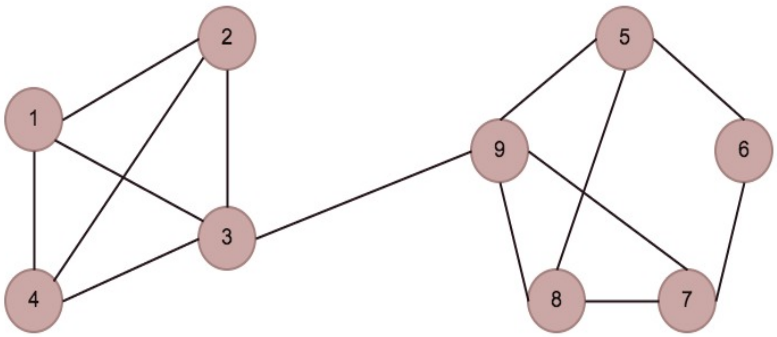


Figure 3: An example network with 9 nodes

3.3 Problem Formulation

3.3.1 Partial Order on Network

In this section, we present several concepts as preliminaries of our Partial Order Preserving (POP) principle for feature selection. Our partial order is defined on an *information network*.

Definition 1 Information Network *An information network $G = (V, E, X)$ consists of V, the group of vertices, $E \subseteq V \times V$, the set of edges, and feature matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ ($i = 1 \ldots n$, $n = |V|$), where $\mathbf{x}_i \in \{0, 1\}^D$ is the attribute vector of node v_i.*

In an information network, for each node v, the remaining nodes can be divided to two categories based on whether they are linked to v: *linked set* and *unlinked set*.

Definition 2 Linked Set *For a node $v \in V$, its linked set is defined as the set $\mathcal{L}(v)$ of all the nodes which are linked to v, i.e., $u \in \mathcal{L}(v) \Leftrightarrow (u, v) \in E$.*

Definition 3 Unlinked Set *For a node $v \in V$, its unlinked set is set of nodes $\mathcal{U}(v)$ which are not in the linked set of v, i.e., $\mathcal{U}(v) = V/\mathcal{L}(v)$*

Traditional i.i.d assumption does not hold for data instances in networks because of the widely observed *homophily effect*. In recent years, many machine learning algorithms on networks try to exploit this fact: *friends are similar*. One popular technique is network based regularization (30) (11), which enforces neighbor nodes (i.e., nodes in linked set) to be similar.

But exploiting information solely from linked sets is not sufficient for feature selection. Though good features are likely to be shared by neighbors, not all features shared by neighbors are of high

quality. For example, in citation network, neighbors (i.e., cited and citing paper) are usually of similar topic because of the homophily effect. As a result, they usually share some topical words (e.g. *SVM*, *LDA*). But indiscriminative words such as *propose* and *compare* are also shared by many neighbors. So we take one step further to exploit both the linked sets and unlinked sets: *friends are usually* **more** *similar than non-friends*. Good features should make neighbors look similar and non-neighbors not so similar. We formulate this idea as link-based partial order as follows.

Definition 4 Link-based Partial Order *We formulate such property as partial order $j >_i k$, where node v_j and node v_k are in the linked set and unlinked set of node v_i, respectively. Node v_i is referred to as the pivot of this partial order. Such partial order is denoted as a triplet (i, j, k) or $j >_i k$.*

$$sim(v_i, v_j) > sim(v_i, v_k), v_j \in \mathcal{L}(v_i), v_k \in \mathcal{U}(v_i) \tag{3.1}$$

Let us take the network with 9 nodes in Figure 3 for example. The linked set $\mathcal{L}(v_3)$ of node v_3 is $\{v_1, v_2, v_4, v_9\}$, while its unlinked set $\mathcal{U}(v_3)$ is $\{v_5, v_6, v_7, v_8\}$. Generally speaking, $\{v_1, v_2, v_4, v_9\}$ should resemble v_3 more than $\{v_5, v_6, v_7, v_8\}$ to v_3. There are $4 \times 4 = 16$ partial order triplets (e.g., $(3, 1, 6)$, $(3, 1, 7)$, $(3, 2, 5)$) w.r.t *pivot* v_3.

This link-based partial order aims to capture the difference between linked set and unlinked set, i.e., what distinguishes linked set from unlinked set. The major difficulty of unsupervised feature selection comes from the lack of label, as the labels can provide clear guidance: features providing good separability of different classes are high-quality ones. In unsupervised scenario, we will show partial order can

TABLE IV: Symbol definitions

Symbol	Definition
$\mathbf{x}_i \in \{0,1\}^D$	Feature vector of node v_i
$\mathcal{L}(v_i)$	Linked set of node v_i
$\mathcal{U}(v_i)$	Unlinked set of node v_i
s_{ij}	Similarity between node v_i and v_j after feature selection
s_{ijk}	Difference between s_{ij} and s_{ik}
$j >_i k$	Partial order triplet in which $v_j \in \mathcal{L}(v_i), v_k \in \mathcal{U}(v_i)$
(i,j,k)	Same as above
Ω	Set of all partial order triplets (i,j,k)
$l(j >_i k)$	The extent to which $j >_i k$ is preserved
$L(>)$	The extent to which all partial orders are preserved
$\mathbf{w} \in \{0,1\}^D$	Feature selection indicator vector

serve a similar purpose as class label. Features of good quality should be able to distinguish the linked set from the unlinked set, which is the intuition underlying our approach.

3.3.2 Partial Order Preserving Feature Selection (POPFS)

Suppose the feature vector of node v_i is $\mathbf{x}_i \in \{0,1\}^D$ and our goal is to select d ($d < D$) features. Without loss of generality, we assume binary features since categorical or numerical features can be transformed to binary features (e.g., by binning). In order to do feature selection, we introduce an indicator vector $\mathbf{w} = (w_1, w_2, \ldots, w_D)^T$, $w_i \in \{0,1\}$ ($\forall i = 1, \ldots, D$). Then we construct a diagonal matrix $diag(\mathbf{w})$ from $\mathbf{w}$. Therefore, the data instance $\mathbf{x}_i$ after feature selection is $diag(\mathbf{w})\mathbf{x}_i$. A set of important symbols used in this chapter are summarized in Table IV.

Based on the link-based partial order defined above, it is desirable that partial order is preserved after feature selection. This can be formulated as follows.

$$\text{sim}(diag(\mathbf{w})\mathbf{x}_i, diag(\mathbf{w})\mathbf{x}_j) > \text{sim}(diag(\mathbf{w})\mathbf{x}_i, diag(\mathbf{w})\mathbf{x}_k) \tag{3.2}$$

In principle, $sim(\cdot,\cdot)$ could be any similarity metric defined on the feature vector, such as Cosine Similarity. To make the optimization simple, we use inner product as the similarity measure. We denote $sim(diag(\mathbf{w})\mathbf{x}_i, diag(\mathbf{w})\mathbf{x}_j)$ as s_{ij}. Rather than the absolute values of s_{ij} and s_{ik}, we are more interested in their relative difference s_{ijk}.

$$\begin{aligned} s_{ijk} &= s_{ij} - s_{ik} \\ &= \mathbf{x}_i^T diag(\mathbf{w})\mathbf{x}_j - \mathbf{x}_i^T diag(\mathbf{w})\mathbf{x}_k \end{aligned} \tag{3.3}$$

We further define an objective function $l(j >_i k \mid \mathbf{w})$ over the partial order triplet (i, j, k) to quantify to what extent the partial order $j >_i k$ is preserved.

$$l(j >_i k \mid \mathbf{w}) = f(s_{ijk} \mid \mathbf{w}) \tag{3.4}$$

A monotonically non-decreasing link function f is used to connect $l(j >_i k)$ with s_{ijk}. When s_{ijk} is large, it means (i, j, k) is well preserved; when s_{ijk} is small (e.g., a negative value), it means (i, j, k) is poorly preserved. Different types of link function can be adopted, for example, identity function or sigmoid function.

However, similar nodes may not be always connected in networks. For example, in co-author network, *Jiawei Han* and *Christos Faloutsos* have not collaborated though they work on similar research topics. So we cannot expect every $(j >_i k)$ derived from the network to be preserved. But in an aggregate sense, a set of good features should make the partial order triplets derived from network structure minimally violated (i.e., maximally preserved). Let us denote the set of all the partial order triplets as Ω.

$$\Omega = \{(i, j, k)|i \in V, j \in \mathcal{L}_i, k \in \mathcal{U}_i\} \tag{3.5}$$

We are interested in preserving the aggregated partial order $L(>)$. This leads to maximizing $l(\cdot)$ over all triplets with constraint $\sum_{i=1}^{D} w_i = d$ where d is the number of selected features.

$$\begin{aligned} \max_{\mathbf{w}} L(>) &= \sum_{(i,j,k)\in\Omega} l(j >_i k \mid \mathbf{w}) \\ &= \sum_{i\in V}\sum_{j\in\mathcal{L}_i}\sum_{k\in\mathcal{U}_i} f(s_{ijk} \mid \mathbf{w}) \\ &\text{s.t. } w_i \in \{0, 1\},\ \sum_{i=1}^{D} w_i = d \end{aligned} \tag{3.6}$$

3.4 Instantiations of the POP Framework

In previous section, we introduce the unified framework for Partial Order Preserving Feature Selection (POPFS). In this section, we present three instantiations of the POP principle: Simple POP, Probabilistic POP and Max-Margin POP, which have different interpretations.

3.4.1 Simple POP (SPOP)

For simplest case of link function, we can use identity function as f. It is easy to show that the optimization problem in Equation 3.6) is equivalent to calculating the following score for each feature.

$$score(a) = \sum_{(i,j,k)\in\Omega} I(i,j,a) - \sum_{(i,j,k)\in\Omega} I(i,k,a) \tag{3.7}$$

where $I(i,j,a)$ is an indicator function, which equals 1 if both nodes i and j have feature a and equals 0 otherwise. The first part of the score is the number of neighbor pairs sharing this feature a, which we refer to as the *linked score* of feature a; the second part of the score is the number of non-neighbor pairs sharing feature a, referred to as *unlinked score*. The final score of each feature is the difference between linked score and unlinked score. After we calculate the score using Equation 3.7, we can simply select the top d features with the highest scores. By using identity link function, it does not consider interaction among features and therefore each feature can be evaluated independently.

This decomposition reveals several useful properties about SPOP and provides better understanding of this principle. If a feature's final score is above zero, it means its linked score is larger than its unlinked score. This indicates that, statistically, this feature appear more often in linked nodes than in non-linked nodes. Consider for example a citation network with papers from several topics (e.g., Machine Learning, Database, System). A generic feature (e.g., stop word) will have both high linked score and unlinked score because of its indiscriminative presence in nodes. The final score will be low as a result. The domain-specific features (e.g., *SVM*, *classification*) tend to have high linked scores and relatively low unlinked scores. Hence, the domain-specific terms will be retained and generic terms

will be discarded by the feature selection process. As a result, unsupervised learning tasks, such as clustering, will benefit from this.

Although real-world networks can provide rich link information for constructing partial orders, they are often noisy by nature. If a noisy link connects two dissimilar nodes by accident, it will have minimal impact on the score calculated by SPOP. For example, given node v_i, consider two nodes $v_j \in \mathcal{L}_i$ and $v_k \in \mathcal{U}_i$. Suppose both v_j and v_k are not similar to v_i but v_j appears in $\mathcal{L}_i$ as noise. For an indiscriminative feature a, v_j and v_k would have similar probability to have it. So, by expectation this will not increase $score(a)$ since $E[I(i,j,a) - I(i,k,a)] \approx 0$. If we only utilize *linked set* through preserving Graph Laplacian without using *unlinked set*, feature selection would be possibly misled by noisy links. This illustrates another strength of preserving partial order against preserving the absolute value of similarity.

3.4.2 Probabilistic POP(PPOP)

Though SPOP is simple and intuitive, it evaluates features individually and hence fails to take into consideration the correlation between features. In this and the following section, we develop two instantiations which evaluate features jointly.

From a generative point of view, we assume all the partial orders are generated from the indicator vector $\mathbf{w} \in \{0,1\}^D$. More specifically, we model the probability of preserving partial order $j >_i k$ as

$$P(j >_i k \mid \mathbf{w}) = \sigma(s_{ijk}) \tag{3.8}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function. The larger s_{ijk} is, the more likely partial order $j >_i k$ is preserved. By assuming the partial orders to be independent, the probability $P(> |w)$ of all the partial orders being respected given $\mathbf{w}$ is,

$$\begin{aligned} P(> |\mathbf{w}) &= \prod_{(i,j,k)\in\Omega} P(j >_i k|\mathbf{w}) \\ &= \prod_{(i,j,k)\in\Omega} \sigma(s_{ijk}) \end{aligned} \tag{3.9}$$

The goal is to find the feature indicator vector $\mathbf{w}$ which maximizes $P(> |\mathbf{w})$ (i.e., to preserve the aggregated partial orders with maximum probability). Learning this model can be performed by maximizing the log-likelihood,

$$\begin{aligned} \max_{\mathbf{w}} \ \log P(> |\mathbf{w}) &= \sum_{(i,j,k)\in\Omega} \log P(j >_i k|\mathbf{w}) \\ &= \sum_{(i,j,k)\in\Omega} \log \sigma(s_{ijk}) \\ &\text{s.t. } w_i \in \{0,1\},\ \sum_{i=1}^{D} w_i = d \end{aligned} \tag{3.10}$$

It provides a probabilistic interpretation for the partial order preserving principle. The connection between Equation 3.17 andEquation 3.6 is easy to see: $\log \sigma(\cdot)$ is used as the link function.

3.4.3 Max Margin POP (MMPOP)

Structured learning methods, such as Structural SVM (39), have gained substantial popularity in the past decade and are powerful for combinatorial optimization. Preserving partial order is to well separate

the linked and unlinked sets for each given pivot, which fits well into structural learning framework as follows.

$$\begin{aligned} \min_{\mathbf{w}} \ & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \ & s_{ijk} \geq 1, \forall (i,j,k) \in \Omega \end{aligned} \tag{3.11}$$

However, in real world networks, the linked set and unlinked set are not always linearly separable using $\mathbf{w}$, as in the *Jiawei Han/Christos Faloutsos* example. So, to address this issue, we add an slack variable μ_{ijk} to impose soft margin.

$$\begin{aligned} \min_{\mathbf{w}} \ & \sum_{(i,j,k)\in\Omega} \mu_{ijk} \\ \text{s.t.} \ & s_{ijk} \geq 1 - \mu_{ijk}, \forall (i,j,k) \in \Omega \\ & w_i \in \{0,1\}, \ \sum_{i=1}^{D} w_i = d \end{aligned} \tag{3.12}$$

To make clear its connection to the Equation 3.6 in the general framework, we rewrite it as follows.

$$\begin{aligned} \max_{\mathbf{w}} \ & \sum_{(i,j,k)\in\Omega} -\max(0, 1 - s_{ijk}) \\ \text{s.t.} \ & w_i \in \{0,1\}, \ \sum_{i=1}^{D} w_i = d \end{aligned} \tag{3.13}$$

So, Equation 3.18 is equivalent to using negative hinge loss as link function in Equation 3.6.

3.4.4 Connection to AUC Optimization

To further justify using the POP principle for feature selection, we show how it is related to optimizing AUC. AUC (Area Under ROC Curve) is a widely used metric for evaluating binary prediction problem such as recommender system and link prediction. Optimizing the objective based on POP optimizes the AUC for link prediction.

$$AUC(v_i) = \frac{1}{|\mathcal{L}_i||\mathcal{U}_i|} \sum_{j \in \mathcal{L}_i} \sum_{k \in \mathcal{U}_i} I(s_{ijk} > 0) \tag{3.14}$$

where indicator function $I(\cdot)$ returns 1 if $s_{ijk} > 0$ and 0 otherwise .

$$\begin{aligned} AUC &= \frac{1}{|V|} \cdot \sum_{i \in V} AUC(v_i) \\ &= \frac{1}{Z} \sum_{(i,j,k) \in \Omega} I(s_{ijk} > 0) \end{aligned} \tag{3.15}$$

where $Z = |\mathcal{L}_i||\mathcal{U}_i||V|$ is a normalizing constant. Comparing the objective function in Equation 3.6)with Equation 3.15, it is obvious to observe the connection with AUC optimization. AUC uses a non-continuous indicator function $I(\cdot)$ as the loss function, while PPOP and MMPOP use continuous loss function (logistic loss and hinge loss, respectively) to approximate the non-continuous counterpart.

Features selected by methods following different principles tend to have different properties. From the analogy between POP and AUC, we know that features selected by POP based methods are optimal in terms of preserving the network structure. This implies that POP-based feature selection methods can be particularly useful for link prediction task.

3.5 Optimization

For Simple POP (SPOP), one only needs to calculate linked score and unlinked score and rank features by their final scores. Optimization for MMPOP and PPOP is a mixed $0-1$ integer programming problem, which is NP-hard in general. To make optimization tractable, we relax the "0/1" constraint in the integer programming problem by replacing $w_i \in \{0, 1\}$ with $w_i \in \mathcal{R}$. Such real-valued weights can be intuitively interpreted as features' *Importance Score*. Then we can rank the features by their importance scores in $\mathbf{w}$ and output the top d features. A challenge for all POP instantiations is that, there are a large number of potential partial order combinations ($O(n|E|)$). It would be very inefficient to iterate through all these $O(n|E|)$ partial order triplets. So we propose to use a bootstrap sampling based technique, *Stochastic (Sub)Gradient Descent*, to solve the optimization problem. In addition to efficiency, sampling based technique is also more robust to noise and outliers.

The objective functions of all three instantiations are convex since they use convex link function f. Since the link functions in SPOP and PPOP are both differentiable, the optimization problem can be efficiently solved by Stochastic Gradient Descent (SGD) method. But MMPOP uses hinge loss which is not differentiable. To solve the optimization problem of MMPOP, we can calculate subgradient and employ Stochastic Subgradient Descent. Hence, all three instantiations can be solved using a unified framework, which is presented in Algorithm 2. In each iteration, we sample a triplet (i, j, k), calculate the (sub)gradient and update $\mathbf{w}$.

Simple POP has the simplest form of gradient.

$$\frac{\partial l(j >_i k)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} s_{ijk} \tag{3.16}$$

For probabilistic POP (PPOP), the gradient for one sample is calculated as follows:

$$\frac{\partial l(j >_i k)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} f(s_{ijk}) \qquad = \frac{e^{-s_{ijk}}}{1 + e^{-s_{ijk}}} \cdot \frac{\partial}{\partial \mathbf{w}} s_{ijk} \tag{3.17}$$

For Max Margin POPFS (MMPOP), we calculate the subgradient and only update the weight vector when $1 - s_{ijk} > 0$:

$$\frac{\partial l(j >_i k)}{\partial \mathbf{w}} = \begin{cases} \frac{\partial}{\partial \mathbf{w}} s_{ijk} & \text{if } s_{ijk} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

For these three approaches,

$$\frac{\partial}{\partial w_p} s_{ijk} = \begin{cases} 1 & \text{if } x_{ip} = 1 \;\&\; x_{jp} = 1 \;\&\; x_{kp} = 0 \\ -1 & \text{if } x_{ip} = 1 \;\&\; x_{jp} = 0 \;\&\; x_{kp} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.19}$$

where x_{ip} is the p-th feature in x_i. From the gradient formula of three approaches, one can observe that the gradient on the p-th feature in SPOP is not influenced by other features. In PPOP and MMPOP, the gradient is impacted by s_{ijk}: when s_{ijk} is large, the gradient is a small value ($e^{-s_{ijk}}/(1 + e^{-s_{ijk}})$) in PPOP or 0 in MMPOP. Such updating scheme addresses the redundancy issue in feature selection.

The optimization error can be bounded as shown in the following theorem.

Algorithm 2 Stochastic (sub)gradient descent algorithm for POP

$\mathbf{w} \leftarrow [0, 0, \ldots, 0]$
for (t in $1..T$) **do**
  step size $\eta_t \leftarrow \frac{1}{\lambda t}$
  update $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t * \Delta_t$, using corresponding formula (Equation 3.16), (Equation 3.17) or (Equation 3.18) for Δ_t
end for
Sort features w.r.t. $w[i]$ and output the top d features

Theorem 3.5.1 *Assume that the data is bounded such that* $max_i\ x_i^T diag(w) x_i\ <\ R$ *and* $R \geq 1$. *In algorithm 2 at iteration* T, *with* $\lambda \leq \frac{1}{4}$, *and batch-size* $B = 1$, $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_T$ *be the average* $\mathbf{w}$ *by iteration* T. *Then, with probability of at least* $1 - \delta$,

$$f(\bar{\mathbf{w}}) - min f(\mathbf{w}^*) \leq \frac{21R^2 ln(T/\delta)}{\lambda T}. \tag{3.20}$$

Proof Sketch: Algorithm 2 is an instance of PEGASOS without a projection step on one-class data. Corollary 2 in (74) proves the same bound for traditional SVM input (without a projection step).

In each iteration, it takes $O(m)$ time to update $\mathbf{w}$, where m is the average number of non-zero features in each data point. This effectively exploits the fact that, in many datasets, m is often small though D can be large. If we sample T triplets of (i, j, k), the overall time complexity is $O(mT)$. Since our goal is feature selection, only the rank of weights w_i is needed. It means $\mathbf{w}$ does not need to be too precise (i.e., δ does not need to be very small). By employing SGD algorithm in Algorithm 2, SPOP, PPOP and MMPOP can be efficiently solved for large-scale networks. In addition, SGD can be updated in an online fashion. This is very useful since new nodes continuously join real-world networks.

TABLE V: Statistics of three datasets

Statistics	Citeseer	Cora	Wiki
# of instances	3312	2708	3363
# of links	4598	5429	33219
# of features	3703	1433	4973
avg. # of non-zero features per instance	31.75	18.17	630.57
# of classes	6	7	19

3.6 Experiment

In this section, we conduct systematic experiments on three publicly available datasets. We compare our POP methods with four baselines on both efficiency and effectiveness. To illustrate how POP methods differ from existing mechanisms, we evaluate the selected features on both clustering task and link prediction task. Experimental results show that POP can select well-rounded features which achieve top performance in both tasks.

3.6.1 Datasets

We use three publicly available network datasets: Citeseer dataset, Cora Dataset and Wikipedia dataset [1] (73). The statistics of three datasets are summarized in Table V.

3.6.2 Baselines

We compared our approach to the following baseline methods.

---

[1]For detailed information about the datasets, one can refer to `http://linqs.cs.umd.edu/projects//projects/lbc/index.html`

- All Features.
- Link Only: Spectral clustering using network links.
- Laplacian Score (LS): Laplacian score (34) selects the features which can best preserve the local manifold structure.
- UDFS: Unsupervised Discriminative Feature Selection (106) is a state-of-the-art pseudo-label based approach for i.i.d data. Unlike Laplacian score, UDFS selects features jointly rather than individually.
- LUFS: Linked Unsupervised Feature Selection is a state-of-the-art unsupervised feature selection method (84) designed for linked social media data, which combines the idea of social dimension (86) with UDFS.

3.6.3 Efficiency

In this section, we investigate the efficiency of POP Feature Selection (POPFS) and the baseline approaches. Baseline methods UDFS and LUFS rely on an iterative method to converge to a local optima. In each iteration, it heavily involves matrix computation and therefore is very inefficient even for a medium-sized ($1000 \sim 10000$) feature set. POPFS has a convex formulation and can be optimized by Stochastic Gradient Decent (SGD). In practice, sampling a small portion of partial order triplets is usually enough. In our experiment, we find sampling $|E| \sim 2|E|$ triplets ($|E|$ is the number of edges) is sufficient for good performance.

Table VI reports the running time of different feature selection algorithms. POPFS requires much less running time than baseline methods (especially UDFS and LUFS). For example, on Citeseer dataset,

TABLE VI: Running time (seconds) of different feature selection algorithms

Dataset	LS	UDFS	LUFS	SPOP	PPOP	MMPOP
Citeseer	10	1234	1420	1	2	2
Cora	5	161	113	1	1	1
Wiki	23	2536	2788	19	22	19

UDFS takes nearly 20 minutes to converge, while POPFS only needs 1 or 2 seconds. The running time of LS is relatively close to POPFS but it only evaluates features individually. Real world social networks (e.g. Facebook and Linkedin) or information networks (e.g., DBLP and biological network) have ever increasing sizes in terms of both number of instances and number of features. Our SGD-based approach can significantly reduce computation time without trading off too much effectiveness.

3.6.4 Results on Clustering

In this section, we evaluate the quality of selected features by their clustering performance. Following the typical setting (106) (84) of evaluation for unsupervised feature selection, we use Accuracy and Normalized Mutual Information (NMI) to evaluate the result of clustering. Accuracy is measured as follows.

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}(c_i = map(p_i)) \tag{3.21}$$

where p_i is the clustering result of data point i and c_i is its ground truth label. $map(\cdot)$ is a permutation mapping function that maps p_i to a class label using Kuhn-Munkres Algorithm.

Normalized Mutual Information (NMI) is calculated as follows. Let C be the set of clusters from the ground truth and C' is obtained from a clustering algorithm.

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))} \tag{3.22}$$

where $H(C)$ and $H(C')$ are the entropy of C and C' and $MI(C, C')$ is the mutual information. Higher value of NMI indicates better quality of clustering.

We use the default parameter setting suggested in the original papers for the baseline methods. For the number of pseudo classes in UDFS and LUFS, we use the ground-truth number of classes. As in previous work (106) (84), we use K-means[1] for evaluation. Since Kmeans is affected by the initial seeds, we repeat the experiment for 20 times and report the average performance. We vary the number of features from 200 to 800, with an increment of 200. The KMeans clustering performance for three datasets is shown in Figure 4.

Among three POP instantiations, MMPOP and PPOP have better clustering performance than SPOP. This demonstrates the importance of evaluating features in a joint manner. SPOP does not take into consideration correlation between features and the redundancy in selected features makes the clustering result suboptimal. With only 200 features, MMPOP and PPOP can obtain much better accuracy and NMI than using all the features. For instance, compared with using all features, MMPOP with 200 features improve the accuracy of KMeans by 10.6% on Citeseer dataset. Besides the improved accuracy and NMI, using selected features rather than all features would also result in speed-up of clustering time.

---

[1]We use the code at `http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html`

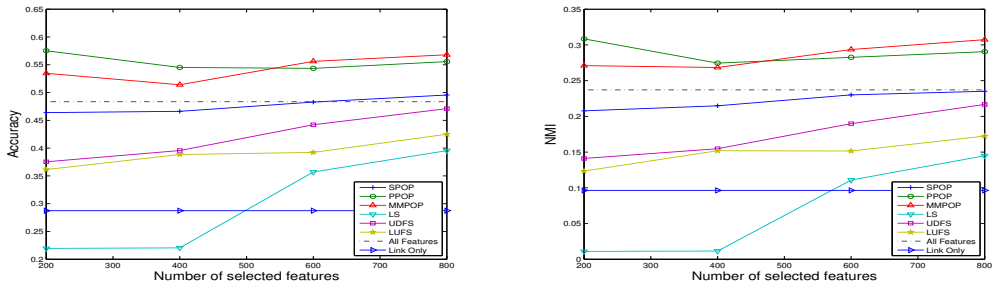
When comparing POP with the baseline methods, we observe that POP based methods (especially PPOP and MMPOP) consistently perform better than baseline methods in terms of both accuracy and NMI. This indicates that POP is an effective criterion for selecting high-quality features. Also, POP tends to obtain good performance with a small number of features (i.e., 200 to 400) while baseline methods normally need more features (i.e., 600 to 800).

Another thing worth noting is the poor performance of clustering with only link structure. Since links in networks are often sparse and noisy, structural information alone is not sufficient to obtain good clusters. But using link structure as guidance to select features achieves much better performance, which illustrates the strength of the POP feature selection. Baseline LUFS exploits link information via extracting social dimensions (86) from links. But social dimensions extracted from noisy and sparse links can be unreliable and this may further mislead the feature selection process.

3.6.5 Partial Order Preserving Property

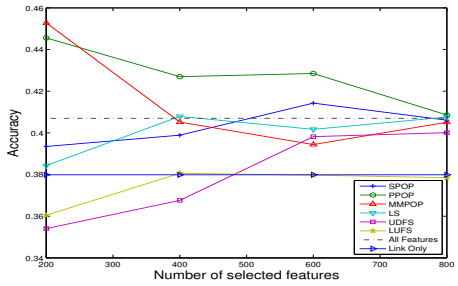
Our approach (POP) has an objective of preserving partial order as described in previous sections. In this section, we illustrate this partial order preserving effect through kNN (we use $k = 1$) link prediction. For each node v, we retrieve the top 1 node u of highest similarity to v. We test if this retrieved node u is an actual neighbor of node v on the network. The precision@1 is shown in Figure 5.

Since this 1NN retrieval uses content only, the prediction performances of all methods are very limited. It also indicates that many similar nodes are not connected in these three datasets. Under such circumstances, POP approaches still outperform other feature selection baselines. This means POP is robust to incomplete link structure.

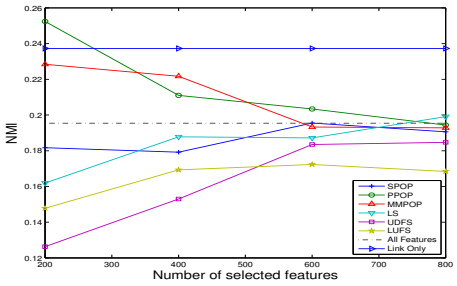


(a) Accuracy on Citeseer

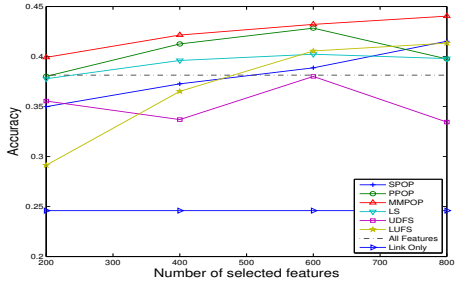
(b) NMI on Citeseer



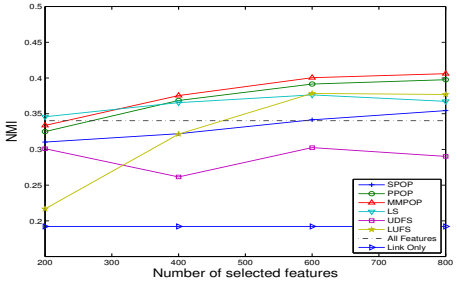
(c) Accuracy on Cora



(d) NMI on Cora



(e) Accuracy on Wiki



(f) NMI on Wiki

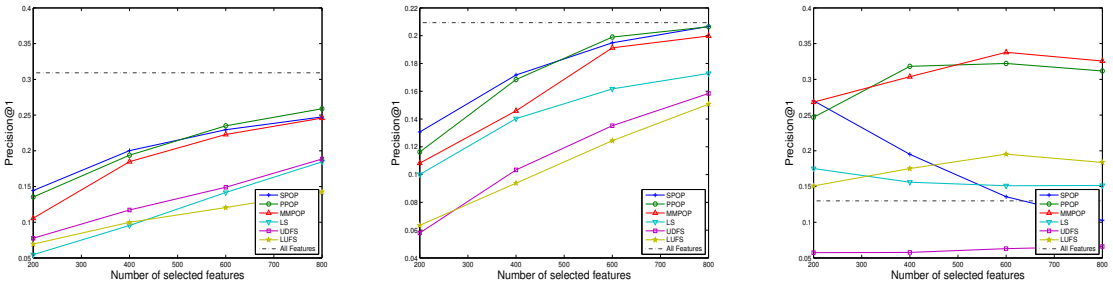
Figure 4: KMeans Results on Three Datasets

POP approaches outperform the baseline methods (LS, UDFS, LUFS) significantly. PPOP and MMPOP usually improve the performance of three other baselines by more than 50% on each dataset. This illustrates that POP's strength in respecting the network structure due to its connection to AUC optimization. The three instantiations of POP perform similarly on Citeseer and Cora datasets. But on Wiki dataset, the performance of SPOP degrades significantly. This is because SPOP ignores the correlation between features and only analyzes each feature individually. This might not result in serious problem when the number of non-zero features in each instance is low (e.g., Citeseer and Cora). However, it would lead to degenerated performance when the number of non-zero features per instance is large, which is the case in Wiki dataset.

LUFS has the ability to incorporate network structure through social dimension. But it utilizes the network information at a community level and fails to exploit the finer grained information of networks. To further understand the difference between different methods, we present the average *document frequency* (df) of features selected by each approach. As shown in Table VII, UDFS tends to select features with high df. This might be fine for clustering, but it loses too much microscopic information. In comparison, PPOP and MMPOP can make a more balanced selection without favoring features with high df in particular. In summary, the features selected by POP are not only better for macroscopic analysis such as clustering, but also good at microscopic analysis because POP respects the local partial order.

TABLE VII: Average document frequency (df) of selected features (top 400)

Dataset	All features	LS	UDFS	LUFS	SPOP	PPOP	MMPOP
Citeseer	28.40	10.23	102.39	76.11	134.30	84.48	70.81
Cora	34.34	52.62	71.61	56.59	80.53	58.42	55.67
Wiki	426.42	598.71	946.91	678.41	1084.40	274.31	262.20



(a) Precision@1 on Citeseer (b) Precision@1 on Cora (c) Precision@1 on Wiki

Figure 5: 1NN Results on Three Datasets

CHAPTER 4

UNSUPERVISED FEATURE SELECTION ON NETWORKS: A GENERATIVE VIEW

(This chapter was previously published as "Unsupervised Feature Selection on Networks: A Generative View", in Proceedings of the 30th international AAAI conference on Artificial Intelligence (AAAI 16), 2016, ©AAAI.)

4.1 Introduction

Network data have become increasingly popular in the past decade, because of the proliferation of various social and information networks. Social networks such as Facebook and Twitter have millions of users all across the world. Different forms of information networks, *e.g.*, co-author networks, citation networks and protein interaction networks, also attract considerable research attention (61) (2). In addition to the link structure, these network data are usually accompanied with content information on the nodes. For example, one can extract thousands of profiling features for users in social networks or ontology features for genes in protein interaction networks. However, redundant and irrelevant features might be included in the high-dimensional feature space. Feature selection (34) (63) is a useful technique since it can help alleviate the curse of dimensionality, speed up the learning process and provide better interpretability. However, not much research effort exists to explore feature selection on networks, especially in unsupervised scenario.

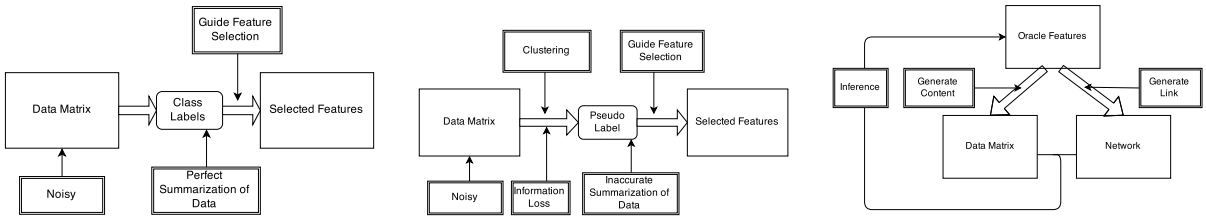
Depending on the availability of class labels, feature selection algorithms can be categorized into supervised methods and unsupervised methods. In the supervised setting, class labels provide a clear guidance to the feature selection process. In the unsupervised setting, feature selection becomes more challenging due to the lack of class labels. In this chapter, we focus on unsupervised feature selection as class labels are usually expensive to obtain. State-of-the-art approaches introduce the notion of pseudo labels (106) (48) (68) to guide the feature selection process. The basic idea is to imitate supervised methods by generating pseudo-labels via certain clustering methods (*e.g.*, spectral clustering and non-negative matrix factorization), and performing sparse regression towards these cluster labels. However, the generated pseudo labels are usually inaccurate and could further mislead the feature selection process.

Moreover, traditional feature selection approaches assume that data instances are independent and identically distributed (i.i.d). In the network data, however, instances are implicitly or explicitly related with certain correlations and dependencies. For example, in research collaboration networks, researchers who collaborate with each other (*i.e.*, connections in the network) tend to share more similar research topics (*i.e.*, close distances in the feature space) than researchers without such collaboration. Most existing feature selection approaches fail to exploit the rich information contained in the links.

Motivated by the importance of feature selection on networks and the deficiency of existing approaches, we propose a novel unsupervised feature selection method from a generative point of view. Our aim is to effectively incorporate information from both link structures and node attributes in the network data. Rather than using potentially inaccurate pseudo labels to guide the feature selection process, we assume that link structures and node attributes are generated by an oracle set of features. We

propose a probabilistic model for this generative process. By performing inference using the linkage and attribute information, we can recover a succinct set of high-quality features. In this manner, we utilize information directly from the network data without generating intermediate pseudo labels. We refer to the proposed approach as Generative Feature Selection (GFS). To our knowledge, no existing method has adopted a generative view for feature selection.

As the state-of-the-art approaches on unsupervised feature selection are mostly pseudo label based methods, we illustrate the essential differences of these approaches and our approach in Figure 6. The class labels can be viewed as a perfect summarization of the data and using them to guide feature selection can usually achieve good performance (6a). Pseudo label based approaches attempt to first summarize the information from the data via clustering, and the pseudo labels serve as a proxy between the original data and the selected features (6b). However, such inaccurate summarization loses much information of the data. Our approach avoids the intermediate step and directly builds connections between the original data and the selected features (*i.e.*, oracle features). As a result, more information from the data could be utilized to guide the feature selection process (6c).



(a) Supervised feature selection (b) Pseudo-label approach (c) Generative Feature Selection

Figure 6: Illustration of different feature selection approaches

4.2 Related Work

4.2.1 Feature Selection for Traditional Data

Feature selection aims to select the most relevant ones from a large number of features and traditional feature selection methods generally fall into three categories: filter models (109) (64), wrapper models (23) and embedded models (17) (87).

Our work focuses on unsupervised scenario as class labels are usually expensive to obtain. One popular guiding principle for unsupervised feature selection is to preserve the local manifold structure or similarity (34) (109) (110). Recently, pseudo label based frameworks (106) (48) (68) have gained much popularity. Unsupervised Discriminative Feature Selection (UDFS) (106) introduces pseudo labels to better capture the discriminative information and the sparsity-inducing $L_{2,1}$ norm is used to select features in an iterative manner. NDFS (48) performs non-negative spectral analysis and feature selection simultaneously. RUFS (68) and RSFS (76) utilizes robust learning framework for generating pseudo labels. Essentially, different pseudo label based methods all use a $L_{2,1}$-regularized regression based framework with different clustering algorithms and constraints on pseudo labels. Since the clustering label is usually far from the ground-truth, it could result in degenerated quality of selected features.

4.2.2 Feature Selection for Network Data

In recent years, efforts have been made towards feature selection on network data. (30) (83) address supervised feature selection on network data via adding network-based regularization term to enforce similarity between neighbors. In unsupervised scenario, POPFS (99) uses network links to guide feature selection efficiently but it fails to use content information. Linked Unsupervised Feature Selection (LUFS) (84) is the only unsupervised feature selection method that utilizes both content and link infor-

mation. LUFS exploits network information through incorporating social dimension based regularization (86) into the UDFS framework (106). It enforces the nodes within the same social dimension to have similar pseudo labels. But the social dimensions generated from links (*e.g.*, by modularity (60) or spectral clustering (62)) and pseudo labels generated from attributes are usually far from accurate, which could mislead the feature selection process.

4.3 Problem Formulation

4.3.1 Preliminaries

In this section, we present several concepts as preliminaries of our unsupervised feature selection method. In the rest of the chapter, we use features and attributes interchangeably. Our goal is to select a set of important features on the network with node attributes, which we refer to as *attributed network*.

Definition 5 (Attributed Network) *An attributed network $G = (V, E, X)$ consists of V, the set of nodes, $E \subseteq V \times V$, the set of links, and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ where $n = |V|$ and $\mathbf{x}_i \in \mathbb{R}^D$ is the feature/attribute vector of the node v_i.*

In the supervised setting, one can select discriminative features that provide good separability of different classes. For unsupervised feature selection, there is no such clear guidance due to the lack of labels. Instead of relying on inaccurate pseudo labels, we aim to directly exploit the information from the data. From a generative point of view, we assume that link structures and node features are generated by an oracle set of features. Our goal is to recover this set of features through inference on the network. Specifically, we assume that there are $d \ll D$ important features among all features which are referred to as *oracle features*. All the node content and the network links are generated by these d oracle features.

We use $\mathbf{s} = \{0, 1\}^D$ as the indicator vector for oracle features, where s_p equals 1 if the p-th feature is an oracle feature and 0 otherwise. Let us denote the diagonal matrix with diagonal elements $\mathbf{s}$ as $\mathrm{diag}(\mathbf{s})$. Therefore, the oracle feature vector of the node v_i is $\mathrm{diag}(\mathbf{s})\mathbf{x}_i$.

4.3.2 Modeling Link Generation

Most unsupervised feature selection methods cannot exploit linkage information. In our generative framework, we can incorporate linkage information seamlessly. From a generative point of view, we assume that the links are generated from a set of oracle features. More specifically, we assume that the probability of a link is determined by the oracle affinity between two nodes defined as follows.

Definition 6 (Oracle Affinity) *Oracle affinity is determined by the dot product of oracle features of two nodes.*

$$a_{ij} = \mathbf{x}_i^T \mathrm{diag}(\mathbf{s})\mathbf{x}_j \tag{4.1}$$

We assume that the oracle affinity is determined by oracle features rather than all the original features to avoid redundancy and irrelevance in the high-dimensional input space. Consider a collection of computer science papers on different topics (*e.g.*, machine learning, operating system, database) and citation links between them. Indiscriminative terms, such as *propose*, *related* and *conclusion*, contain little information in determining the essential similarity between two papers. Since two linked papers are more likely to share similar topics than two random papers, informative terms such as *LDA*, *classification* and *database* would be useful in generating the links. Thus, if a feature is highly indicative of the existence of links, it is likely to be an informative and discriminative feature. By recovering the oracle

features via exploiting network links, we are able to select a set of discriminative features. To achieve this, we introduce the following generative process:

$$\begin{aligned} p_{ij} =& F_g(a_{ij}) \\ E_{ij} \sim& \text{Bernoulli}(p_{ij}) \end{aligned} \tag{4.2}$$

where $F_g(\cdot)$ is a function that transforms the oracle affinity a_{ij} to the linkage probability p_{ij}. $F_g(\cdot)$ should be non-decreasing so that a larger affinity would lead to a larger probability of connection. For example, it could be the sigmoid function, *i.e.*, $F_g(a_{ij}) = 1/(1 + e^{-a_{ij}})$. We further introduce a bias term $b \in \mathbb{R}$, so $F_g(a_{ij}) = 1/(1 + e^{-(a_{ij}+b)})$.

Equation 4.2 describes the generative process from oracle features to the links in networks. By assuming links are i.i.d, the probability of the whole network given the oracle features is as follows:

$$P(G|\mathbf{s}) = \prod_{(i,j)\in E} p_{ij} \cdot \prod_{(i,j)\notin E} (1 - p_{ij}) \tag{4.3}$$

The negative log-likelihood for generating the network links using $F_g(a_{ij}) = \frac{1}{1+e^{-a_{ij}-b}}$ is the following:

$$\begin{aligned}
\mathcal{L}_G &= -\log(P(G|\mathbf{s})) \\
&= -\sum_{(i,j)\in E} \log \frac{1}{1+\exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b\right)} \\
&\quad -\sum_{(i,j)\notin E} \log \frac{\exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b\right)}{1+\exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b\right)} \\
&= \sum_{(i,j)\in V\times V} \log(1+\exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b\right)) \\
&\quad + \sum_{(i,j)\notin E} (\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j + b)
\end{aligned} \tag{4.4}$$

In real-world applications, network data can be very sparse, *i.e.*, linked node pairs are far less than non-linked node pairs. Due to such imbalanced distribution, $\mathcal{L}_G$ would be dominated by the loss on non-linked node pairs. To address this issue, we under-sample the non-linked node pairs to make their size comparable to the linked node pairs. With down-sampling, $\mathcal{L}_G$ is reformulated as follows:

$$\begin{aligned}
\mathcal{L}_G &= -\sum_{(i,j)\in E} \log \frac{1}{1+\exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b\right)} \\
&\quad -\sum_{(i,j)\in SN} \log \frac{\exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b\right)}{1+\exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b\right)} \\
&= \sum_{(i,j)\in E\cup SN} \log(1+\exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b\right)) \\
&\quad + \sum_{(i,j)\in SN} (\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j + b)
\end{aligned} \tag{4.5}$$

where SN denotes the set of sampled non-linked node pairs.

It is worth noting that our link generation approach differs from graph regularization (30) (83) in two important aspects: first, graph regularization usually enforces similarity on linked pairs but fails to utilize information from unlinked pairs; second, graph regularization is usually used on cluster membership/latent factors (as in existing pseudo-label methods) rather than directly on the oracle features. And actually, applying graph regularization on $\mathrm{diag}(\mathbf{s})\mathbf{x}_i$ directly will favor those features that appear indiscriminatively since it fails to penalize features that are frequently shared by unlinked pairs.

4.3.3 Modeling Content Generation

In addition to the linkage information, it is critical to incorporate information from the node content. We assume that each node generates its attributes from the set of oracle features with a mapping function. That is to say, the oracle features can be regarded as a succinct summary of all the features. This intuition can be formalized as follows:

$$\begin{aligned} \boldsymbol{\mu}_i =& F_c(\mathrm{diag}(\mathbf{s})\mathbf{x}_i) \\ \mathbf{x}_i \sim& \mathcal{N}(\boldsymbol{\mu}_i, \sigma^2\mathbf{I}_D) \end{aligned} \tag{4.6}$$

where $\mathcal{N}$ is the Gaussian distribution and $F_c(\cdot)$ is the function that generates $\mathbf{x}_i$ from the oracle features $\mathrm{diag}(\mathbf{s})\mathbf{x}_i$. There could be different choices for the generation function $F_c(\cdot)$. For simplicity, we use a linear mapping as the generating function.

$$F_c(\mathrm{diag}(\mathbf{s})\mathbf{x}_i) = \mathbf{W}^T \mathrm{diag}(\mathbf{s})\mathbf{x}_i \tag{4.7}$$

where $\mathbf{W} \in \mathbb{R}^{D \times D}$ is a projection matrix that represents all the features using oracle features. Only d rows of $\mathbf{W}$ are non-zero which correspond to the non-zero elements of $\mathbf{s}$. If all the original features could be approximated by the oracle features through $F_c(\cdot)$, the oracle features arguably contain the essential information of the node content.

It is easy to verify that, given fixed $\mathbf{W}$, maximizing the log-likelihood of content generation under Equation 4.6 is equivalent to minimizing the sum of square error:

$$||\mathbf{X}^T \operatorname{diag}(\mathbf{s})\mathbf{W} - \mathbf{X}^T||_F^2 \tag{4.8}$$

where $||\cdot||_F$ denotes the Frobenius norm of a matrix. By finding the oracle features that minimize Equation 4.8, we select the most important features that preserve the information of node attributes in the network data. We also need to impose a norm on $\mathbf{W}$ to control its complexity and avoid overfitting. We choose Frobenius norm for the simplicity of optimization.

$$\mathcal{L}_C = ||\mathbf{X}^T \operatorname{diag}(\mathbf{s})\mathbf{W} - \mathbf{X}^T||_F^2 + \beta||\mathbf{W}||_F^2 \tag{4.9}$$

Note that other distributions could also be used for modeling feature generation. For example, one can consider Bernoulli distribution if the features are binary.

$$\begin{aligned} \boldsymbol{\mu}_i =& \frac{1}{1 + \exp(-F_c(\operatorname{diag}(\mathbf{s})\mathbf{x}_i))} \\ \mathbf{x}_i \sim& \text{Bernoulli}(\boldsymbol{\mu}_i) \end{aligned} \tag{4.10}$$

where $\boldsymbol{\mu}_i$ determines the probability of occurrence of $\mathbf{x}_i$. It is easy to see that both Equation 4.6 and Equation 4.10 are special cases of *Generalized Linear Model* (GLM) with different *link functions*. Equation 4.6 corresponds to linear regression and Equation 4.10 corresponds to logistic regression.

4.3.4 Combining Things Together

We have discussed how to generate attributes and links from oracle features in previous sections. Now we put things together and aim to select a set of high-quality features that are optimal considering both content and link generation. Therefore, we aim to minimize the negative log-likelihood on both link and content. By assuming the conditional independence of G and C given b, $\mathbf{s}$ and $\mathbf{W}$, the total negative log-likelihood is as follows:

$$\begin{aligned} \min_{\mathbf{s},b,\mathbf{W}} \quad & -\log P(G, C|\mathbf{s}, b, \mathbf{W}) = \mathcal{L}_G + \mathcal{L}_C \\ \text{s.t.} \quad & s_p \in \{0, 1\}, \forall p = 1, \ldots, D \\ & \sum_{p=1}^{D} s_p = d \end{aligned} \tag{4.11}$$

4.4 Optimization

In this section, we develop a method for performing inference with features and links. The optimization problem in Equation 4.11 is a '0/1' integer programming problem. To make the optimization

tractable, we relax the $0/1$ constraint on $\mathbf{s}$ and only require $\mathbf{s}$ to be a real-valued vector in the range of $[0, 1]$. Moreover, we can write the summation constraint $\sum_{p=1}^{D} s_p = d$ in the form of Lagrangian:

$$\begin{aligned} \min_{\mathbf{s},b,\mathbf{W}} \quad & \mathcal{L} = \mathcal{L}_G + \mathcal{L}_C + \lambda||\mathbf{s}||_1 \\ \text{s.t.} \quad & 0 \leq s_p \leq 1, \forall p = 1, \ldots, D \end{aligned} \tag{4.12}$$

where the sparsity-inducing L_1 norm $||\mathbf{s}||_1$ is equal to $\sum_{p=1}^{D} s_p$, because we enforce $\mathbf{s}$ to be non-negative (*i.e.*, $0 \leq s_p \leq 1$). The value of s_p can be interpreted as the p-th feature's importance score in generating the content and linkage information. Important features would have scores close to 1 and scores of less useful features tend to shrink towards 0. After obtaining the relaxed solution on $\mathbf{s}$, we can rank all the features by their importance scores and select the top d features as the oracle features.

For the optimization problem in Equation 4.12, we need to optimize jointly on the selection vector $\mathbf{s}$, bias term b and the projection matrix $\mathbf{W}$. Since Equation 4.12 is not jointly convex on $\mathbf{s}$, b and $\mathbf{W}$, we adopt an alternating optimization framework to solve it.

Step 1. Fix $\mathbf{W}$ and optimize Equation 4.12 over $\mathbf{s}$ and b.

With fixed $\mathbf{W}$, Equation 4.12 is a convex optimization problem on $\mathbf{s}$ and b. For real-valued $\mathbf{s}$, both $\mathcal{L}_G$ and $\mathcal{L}_C$ is differentiable. For the loss incurred on link structures, the gradient of $\mathcal{L}_G$ with respect to $\mathbf{s}$ can be calculated as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_G}{\partial s_p} = & - \sum_{(i,j) \in E \cup SN} x_{ip}^T x_{jp} \frac{\exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s}) \mathbf{x}_j - b\right)}{1 + \exp\left(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s}) \mathbf{x}_j - b\right)} \\ & + \sum_{(i,j) \in SN} x_{ip}^T x_{jp} \end{aligned} \tag{4.13}$$

The gradient of $\mathcal{L}_C$ with respect to $\mathbf{s}$ is the following:

$$\frac{\partial \mathcal{L}_C}{\partial s_p} = [\mathbf{X}(\mathbf{X}^T \operatorname{diag}(\mathbf{s})\mathbf{W} - \mathbf{X}^T)\mathbf{W}^T]_{pp} \tag{4.14}$$

where $[\cdot]_{pp}$ denotes the p-th diagonal element of matrix $[\cdot]$.

The L_1 norm in general is non-smooth at zero. However, since in our case $\mathbf{s}$ is guaranteed to be non-negative, the L_1 regularization on non-negative $\mathbf{s}$ is differentiable with gradient 1. So the gradient of the whole objective function is

$$\frac{\partial \mathcal{L}}{\partial s_p} = \frac{\partial \mathcal{L}_G}{\partial s_p} + \frac{\partial \mathcal{L}_C}{\partial s_p} + \lambda \tag{4.15}$$

Since we also require $\mathbf{s}$ to be in the range $[0, 1]$, we perform Projected Gradient Descent (PGD) (13) for this constrained optimization problem. We project $\mathbf{s}$ back to $[0, 1]$ after each gradient updating step.

$$\text{Proj}_{[0,1]}(s_p) = \min(\max(0, s_p), 1), \forall p = 1, \ldots, D \tag{4.16}$$

Moreover, the gradient with respect to the bias term b is

$$\frac{\partial \mathcal{L}_G}{\partial b} = -\sum_{(i,j)\in E \cup SN} \frac{\exp(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b)}{1 + \exp(-\mathbf{x}_i^T \operatorname{diag}(\mathbf{s})\mathbf{x}_j - b)} + |SN| \tag{4.17}$$

where $|SN|$ denotes the total number of sampled non-linked node pairs.

Step 2. Fix $\mathbf{s}$ and b, optimize Equation 4.12 over $\mathbf{W}$.

With fixed $\mathbf{s}$, the optimization with respect to $\mathbf{W}$ is convex and we can obtain the closed form solution for $\mathbf{W}$ as follows:

$$\mathbf{W} = (\mathrm{diag}(\mathbf{s})\mathbf{X}\mathbf{X}^T\,\mathrm{diag}(\mathbf{s}) + \beta\mathbf{I}_D)^{-1}\,\mathrm{diag}(\mathbf{s})\mathbf{X}\mathbf{X}^T \tag{4.18}$$

where $\mathbf{I}_D$ is an $D \times D$ identity matrix. Algorithm 3 shows the optimization method based on projected gradient descent. We alternatively perform step 1 and step 2 in an iterative manner until it converges or reaches user-specified maximum number of iterations.

The objective function in Equation 4.12 monotonically decreases in each iteration and it has a lower bound. Hence, Algorithm 3 can converge.

Algorithm 3 Alternating Optimization with Projected Gradient Descent

Initialize: $\mathbf{s}^0 = 0^D$, $b^0 = 0$, $\mathbf{W}^0 = 0^{D\times D}$, $t = 0$.
repeat
$t = t + 1$
Update $\mathbf{s}^t$ and b^t through performing projected gradient descent by Equation 4.15 and Equation 4.17 with $\mathbf{W}^{(t-1)}$
Find the optimal $\mathbf{W}^t$ by Equation 4.18 with $\mathbf{s}^t$.
until converged or $t = maxIterations$
Sort features *w.r.t.* $\mathbf{s}^t$ and output the top d features.

TABLE VIII: Statistics of three datasets

Statistics	Citeseer	Cora	Wiki
# of instances	3312	2708	3363
# of links	4598	5429	33219
# of features	3703	1433	4973
# of classes	6	7	19

4.5 Experiment

In this section, we evaluate the feature quality by performing clustering (community detection) on the features. Experimental results show that GFS significantly outperforms the state-of-the-art methods in terms of feature quality.

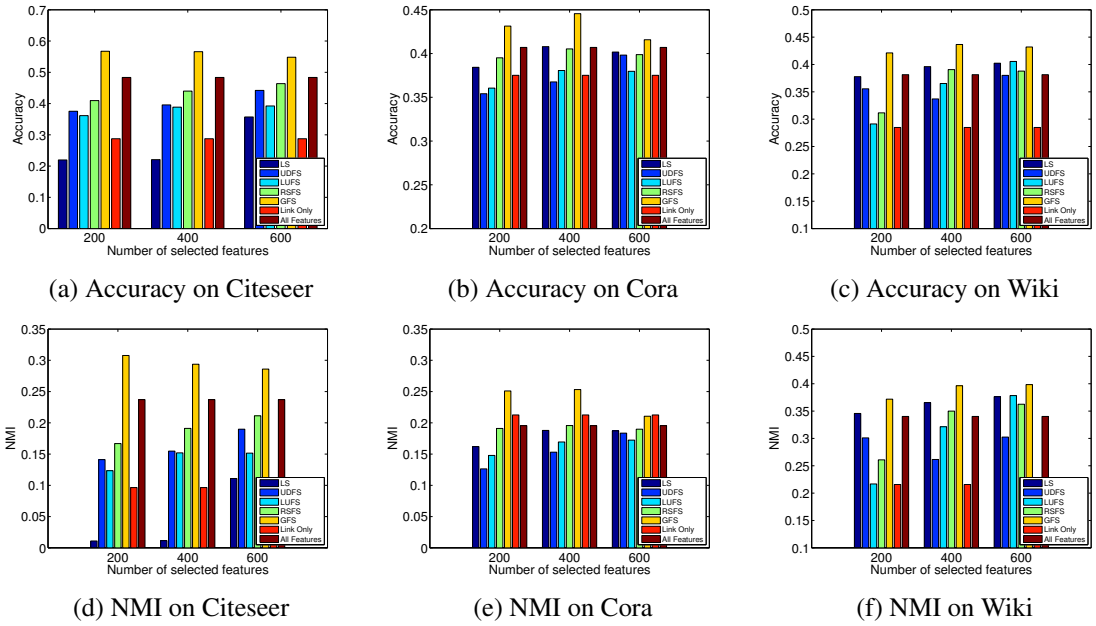
4.5.1 Experiment Setup

We use three publicly available network datasets with node attributes: Citeseer dataset, Cora Dataset and Wikipedia dataset [1] (73). One can refer to the link in the footnote for more details on the datasets. The statistics of three datasets are summarized in Table VIII.

We compared our approach to the following baseline methods: (a) All Features; (b) Link Only (Spectral clustering using network links); (c) LS (Laplacian Score) (34); (d) UDFS (content only) (106) (e) LUFS (which incorporates both content and link information) (84); (f) RSFS (content only) (76).

---

[1] `http://linqs.cs.umd.edu/projects//projects/lbc/index.html`



(a) Accuracy on Citeseer (b) Accuracy on Cora (c) Accuracy on Wiki

(d) NMI on Citeseer (e) NMI on Cora (f) NMI on Wiki

Figure 7: Clustering results on three datasets

Following the typical setting (106) (84) of evaluation for unsupervised feature selection, we use Accuracy and Normalized Mutual Information (NMI) to evaluate the result of clustering. Accuracy is measured as follows.

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}(c_i = map(p_i)) \tag{4.19}$$

where p_i is the clustering result of data point i and c_i is its ground truth label. $map(\cdot)$ is a permutation mapping function that maps p_i to a class label using Kuhn-Munkres Algorithm.

NMI is calculated as follows. Let C be the set of clusters from the ground truth and C' is obtained from a clustering algorithm.

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))} \tag{4.20}$$

where $H(C)$ and $H(C')$ are the entropy of C and C' and $MI(C, C')$ is the mutual information. Higher value of NMI indicates better quality of clustering.

Since it is difficult to determine the optimal values of parameters in unsupervised setting, we use the parameter setting for the baseline methods as suggested in the sensitivity analysis section of the original papers. For the number of pseudo classes in UDFS, LUFS and RSFS, we use the ground-truth number of classes. For the proposed method GFS, we found it is not sensitive to the parameters in a reasonable range. So we fix the parameters of GFS for all datasets with $\beta = 1$ and $\lambda = 1$.

As in previous work (106) (84), we use K-means[1] for evaluation. Since K-means is affected by the initial seeds, we repeat the experiment for 20 times and report the average performance. We vary the number of features in the range $\{200, 400, 600\}$. The K-means clustering performance for three datasets is shown in Figure 7.

4.5.2 Results

We can observe from Figure 7 that feature selection is an effective way to enhance the clustering/community detection performance. With much less features, GFS can obtain significantly better accuracy and NMI than using all the features. For instance, compared with using all features, GFS

[1]We use the code at `http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html`

with 200 features improves the accuracy of clustering by 21.0%, 6.0% and 10.4% on Citeseer, Cora and Wikipedia, respectively. This illustrates the importance of feature selection on networks, since the original feature space can have many low quality/noisy features. It is also worth noting that clustering using only links does not perform very well. This is because network links are often sparse and noisy, and structural information alone is not sufficient to obtain good clusters. But using link structures as guidance in addition to the node content to select features can achieve much better performance, which illustrates the strength of our proposed GFS framework.

When comparing GFS with other unsupervised feature selection approaches, we observe that GFS performs consistently better than baseline methods on different datasets with different numbers of selected features. This indicates that the proposed generative view is an effective framework for selecting high-quality features on network data. LS, UDFS and RSFS are unable to exploit network structure and do not perform as well as GFS. Compared with the most competitive feature selection baseline RSFS, GFS outperforms RSFS by 44.5%, 9.2% and 35.2% with 200 features on three datasets, respectively. Baseline LUFS also attempts to exploit link information via extracting social dimensions (86) from links. But social dimensions extracted from noisy and sparse links can be unreliable and this may further mislead the feature selection process. For example, in Citeseer dataset, the network is sparse and each node only has 1.39 links on average. So the derived social dimensions make LUFS even worse than UDFS and RSFS which do not utilize linkage information. In contrast, GFS can benefit from exploiting the links even when the network structure is sparse, as shown in the case of Citeseer dataset.

In summary, noisy features can be detrimental to the performance of clustering/community detection and appropriately designed unsupervised feature selection method, such as GFS, can alleviate this issue.

CHAPTER 5

LEARNING REPRESENTATION CONSENSUS WITH COUPLED FEATURE SELECTION FOR CROSS VIEW LINK PREDICTION

(This chapter was previously published as "Cross View Link Prediction by Learning Noise-resilient Representation Consensus" in Proceedings of the 26th International World Wide Web Conference (WWW 17), with the permission to reuse from 2017 International World Wide Web Conference Committee)

5.1 Introduction

The optimality of feature selection can depend upon the specific application. For example, features that are optimal for community detection/clustering might not be optimal for link prediction. In this chapter, we use the cross view link prediction problem to illustrate how to formulate the feature selection problem based on the application objective. Specifically, representation learning is performed jointly with feature selection and they could be mutually enhanced for link prediction task.

In the past decade, there have been an increasing number of information networks from a wide range of domains. Study on computer networks, biological and social networks has attracted great attention from the research community (33) (10) (96). Link prediction (1) (2), which aims at recommending potential links between network nodes, is an important step to understand and study the characteristics of these networks. For instance, in bioinformatics, by predicting protein interaction links, one does not need to conduct expensive experiments on all possible pairs and can spend the resource wisely on the most likely interaction. For social media websites, such as Facebook and Twitter, it is fundamen-

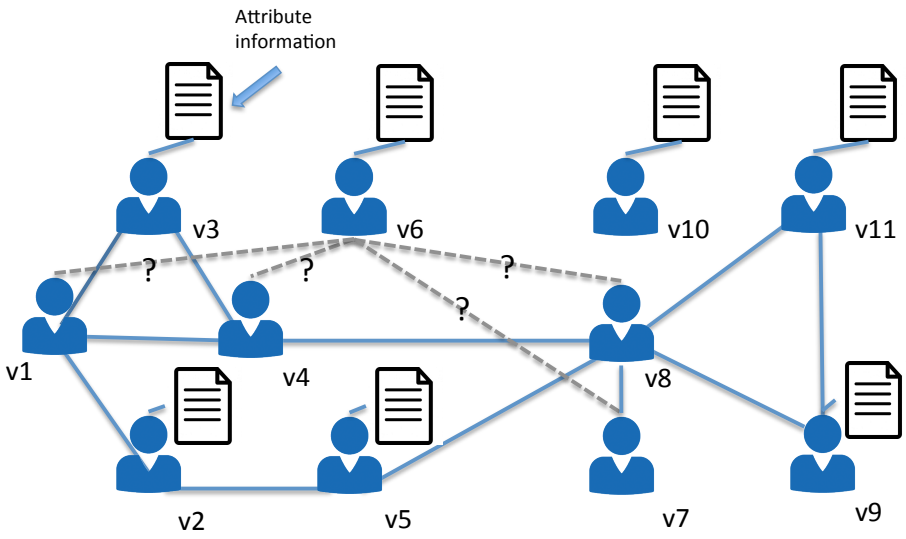


Figure 8: An example of networks with partially observable links and attributes

tal to grow the user base and enhance user engagement with link prediction techniques. For security analysts/agencies, predicting (currently unobserved) links can reveal hidden but important relationship among terrorists and provides additional insights for understanding organizational structures of terrorist-attack activities.

Many methods have been proposed for the task of link prediction (49) (1) (41) (2). However, in various social and information networks, it is common that certain nodes do not have any link information revealed (94) and make these methods not applicable:

- In real-world social networks (e.g., Facebook and Twitter), link prediction for new users usually has the challenge of cold start problem, since these users do not have any connection. Besides, some users may choose a strict privacy setting that restricts the visibility of their connections,

personal information or posts[1][2]. Recommending links in such a partially observable setting could enhance user experience.

- In bioinformatics information networks, for example, studying protein interaction could help researchers better understand many biological processes. However, it is infeasible to collect all the experimental data for all the possible pairs of protein.
- In terrorist-attack networks, nodes represent terrorist activities and links represent terrorist attacks in which the same terrorist group is involved. Detecting hidden links in these networks is useful for understanding the underlying structure of terrorist-attack activities. However, the complete linkages between attacks are highly difficult to resolve (51).

Nonetheless, nodes in many social/information networks are often equipped with features/attributes, such as user attributes in social networks, paper content in co-authorship networks and gene properties in biological networks. These node attributes can help when the link information is not observable. For example, for a new user who joins a social network with few connections (i.e., links), we can utilize his/her user profile (i.e., node attributes) filled out in the registration process to suggest potential links to such a new user, based on the profile similarity.

However, due to the difficulty in data collection, the node attributes of real-world networks also tend to be partially observable in a variety of scenarios and this poses additional challenges for link prediction.

---

[1] https://www.facebook.com/help/325807937506242/

[2] https://help.linkedin.com/app/answers/detail/a_id/52

- In online social networks, some users might not fill up profile information when registering or have not yet started to write posts. Besides, a user might choose a privacy level with which no one or only friends could view his/her posts and profile information.
- For information networks in domain of bioinformatics, it can be costly to obtain features for certain genes or proteins.
- In terrorist network, the difficulty of collecting attributes/profiles for different terrorists varies. For example, the information of terrorists with higher ranks is often protected better than that of an ordinary terrorist. Also, it is usually difficult to obtain all the necessary attributes for a newly joined terrorist.

Hence, for real-world networks, assuming *partially observable networks* is a more realistic setting, in which only a certain fraction of nodes have both connections and node attributes, whereas the other nodes have either links or attributes unobservable. Consider the example in Figure 8. The network has 5 nodes with both link and attribute information and other nodes are partially observable. While the link information of node v_6 is missing, we could recommend potential friends from the candidate pool $\{v_2, v_3, v_5, v_9, v_{10}, v_{11}\}$ based on their attribute similarity. However, it would be more challenging to recommend from the candidates $\{v_1, v_4, v_7, v_8\}$ which only have link information. We refer to such problem as **Cross View Link Prediction (CVLP)**, in which we recommend nodes with only attributes to nodes with only links (or vice versa).

The CVLP task can be even more challenging in many real-world social/information networks, as node attributes are usually characterized by high dimensionality and contain certain amount of noisy/irrelevant attributes. For example, in Facebook network, one could extract millions of (sparse)

features for user profiling, such as the groups a user has joined, the web pages he has liked, the content of posts, and the user's demographic features. Such high-dimensional features pose additional challenges to link prediction task. These features have different importances in predicting the links and some features might even have negative effect on the prediction. So it is critical to select only the relevant features for link prediction.

In this chapter, we study the novel problem of CVLP, and propose an effective approach, Noise-resilient Representation Consensus Learning (NRCL), to address these challenges of cross view link prediction. Since nodes with only links and nodes with only attributes are not directly comparable in their original form, we propose to learn a common subspace in which nodes with incomplete information become comparable to each other. We utilize link-based representations and content-based representations of fully observable nodes to form a co-regularization consensus. Experimental results on real-world datasets demonstrate that NRCL outperforms baseline methods significantly. The contribution of the chapter can be summarized as follows:

- To our best knowledge, we are the first to formulate and investigate the problem of cross-view link prediction on networks with partially observable links and node attributes.
- We propose to learn representation consensus so that nodes with either link information or node attributes could become comparable in the latent space. Two instantiations of the proposed framework, based on log loss and Huber loss, are developed and compared, with the latter being more robust to noisy link structure.

- Considering that many node attributes in real-world networks tend to be noisy/irrelevant, we perform joint feature selection in our framework to alleviate the issue of noisy attributes. To our knowledge, no prior work on node representation learning selects features jointly.
- We conduct experiments on four real-world networks and show the effectiveness of the proposed method on the task of cross-view link prediction.

5.2 Related Work

The link prediction problem has been studied extensively by researchers from the machine learning and data mining community (1) (107) (103). Various scoring methods have been proposed based on the topology of graphs: 1) Common Neighbor based methods: Adamic/Adar (1) assigns weight to each common neighbor based on the degree of the neighbors; 2) Path based methods such as Katz (41) and Local Path and Random Walk with Restart (54). Katz (41) is a path based method which sums over all paths between two nodes.

Some link prediction methods (50) (2) formulate link prediction as a supervised task where the existence of link is used as supervision. For example, Lichtenwalter et al. studied how to ensemble different measures for link prediction (50). Supervised Random Walk (2) is a random walk based approach to combine different similarity scores. It attempts to learn a weight for different features to make the transition probability between linked nodes larger than that of unlinked nodes.

Some work investigates the low rank approximation methods by generating a low rank matrix to approximate the adjacency matrix of network structure (91) (56). Besides, various latent variable models (3) (107) (58) have been proposed to model the relationship between nodes. For example, WTFW (3), a topic model-based approach, can perform link prediction as well as providing explanation to support

the prediction. Recently, embedding methods, such as DeepWalk (65), LINE (81) and node2vec (29), are developed to learn representations for network nodes based on the link structure. They employ similar objective function as the popular word embedding method Word2Vec (57) and the derived node embedding can be used for link prediction (29).

Recently, researchers study how to perform link prediction for the heterogeneous information network (79) (108) (103), where multiple types of nodes and links exist in the network.

However, existing methods usually assume the network structure is complete. No previous research studies cross-view link prediction on partially observable networks.

5.3 Formulations

In this section, we present a few preliminary definitions that will be used in the rest of this chapter.

Definition 7 Information Network *An information network $G = (V, E, X)$ consists of V, the set of nodes, $E \subseteq V \times V$, the set of edges, and a feature matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ ($n = |V|$), where $\mathbf{x}_i \in R^D$ ($i = 1 \ldots n$) is the attribute vector of node v_i.*

Since various real-world social/information networks have partially observable links and node attributes, we study link prediction on such networks defined as follows.

Definition 8 Partially Observable Information Network *In a partially observable information network $G = (V, E, X)$, each node can belong to one or two of the following (overlapping) sets: the set of nodes O^g with observable links and nodes O^a with observable attributes. We also use $O^s = O^g \cap O^a$ to denote the set of nodes with both observable links and attributes .*

Note that $V = O^g \cup O^a$ since we assume each node has at least one source of information. In real world, certain nodes might have disclosed neither links nor attributes. We do not consider such nodes since no information can be used to suggest link to them in that case. In a partially observable information network $G = (V, E, X)$, many nodes have only one view of information, i.e., link or node attributes. We refer to nodes with only links ($O^g \backslash O^s$) as **link-only nodes** and nodes with only attributes ($O^a \backslash O^s$) as **attribute-only nodes**. In this chapter, we study how to recommend link-only nodes to attribute-only nodes, or vice versa. We refer to such task as **Cross View Link Prediction (CVLP)**.

Let the number of nodes with links, nodes with attributes and nodes with both links and attributes be $n^g = |O^g|$, $n^a = |O^a|$ and $n^s = |O^s|$, respectively.

5.4 Link-based Representation Learning

We aim to learn representations for the network nodes by preserving structural information. For a node v_i, other nodes can be divided into to two classes, neighbors and non-neighbors. Hence, we can derive triplets (i, j, k) from the network structure, where v_i and v_j are neighbors while v_i and v_k are non-neighbors. We denote the set of all such triplets (i, j, k) as Ω.

Let us denote the representation learned from links as $\mathbf{U}^g \in \mathbb{R}^{n^g \times m}$, where m is the number of dimensions in the representation. The affinity s_{ij} between two nodes v_i and v_j can be calculated as the inner product of the representations $s_{ij} = \mathbf{U}^g_i (\mathbf{U}^g_j)^T$. To make the representation appropriate for

link prediction, it is desirable to make the affinity between neighbors larger than the affinity between non-neighbors. So we aim to optimize the following objective.

$$\begin{aligned} &\min_{\mathbf{U}^g} \ ||\mathbf{U}^g||_F^2 \\ &\text{s.t. } s_{ij} \geq s_{ik}, \forall (i,j,k) \in \Omega \end{aligned} \tag{5.1}$$

This objective function minimizes the complexity of representation while keeping neighbors and non-neighbors separable. Since it might not be possible to satisfy all the hard constraint on all triplets (i, j, k), we minimize the number of mis-ordered ranking triplets. Let us denote $s_{ijk} = s_{ij} - s_{ik}$ and the objective function is the following.

$$\min_{\mathbf{U}^g} \sum_{(i,j,k)\in\Omega} \mathbf{I}(s_{ijk} < 0) + \lambda_g ||\mathbf{U}^g||_F^2 \tag{5.2}$$

where $\mathbf{I}(\cdot)$ is an indicator function which returns 1 if $(\cdot)$ is true and 0 otherwise. The 0/1 loss function is not smooth and is computationally intractable to optimize. So we replace it with a continuous convex surrogate loss $l(\cdot)$ in the objective function.

$$\min_{\mathbf{U}^g} \sum_{(i,j,k)\in\Omega} l(s_{ijk}) + \lambda_g ||\mathbf{U}^g||_F^2 \tag{5.3}$$

AUC (Area Under ROC Curve) is a widely used metric for evaluating binary prediction problem such as recommender system and link prediction (50). It can be shown that optimizing the objective in Equation 5.2 is related to optimizing the AUC (70) (99). Hence, learning the representation under such objective is a good choice for link prediction. There can be different options for the loss function

$l(s_{ijk})$, such as log-loss, exponential loss and hinge loss. In the following subsection, we develop two instantiations of NRCL with different loss functions.

5.4.1 Probabilistic Representation Learning (P-RL)

From a generative point of view, one can assume all the triplets $(i, j, k) \in \Omega$ are generated from the node representation $\mathbf{U}^g$. More specifically, we model the probability of preserving ranking order $s_{ij} > s_{ik}$ using the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$.

$$P(s_{ij} > s_{ik} \mid \mathbf{U}^g) = \sigma(s_{ijk}) \tag{5.4}$$

The larger s_{ijk} is, the more likely ranking order $s_{ij} > s_{ik}$ is preserved. By assuming the ranking orders to be independent, the probability $P(> |\mathbf{U}^g)$ of all the ranking orders being preserved given $\mathbf{U}^g$ is the following.

$$\begin{aligned} P(> |\mathbf{U}^g) &= \prod_{(i,j,k)\in\Omega} P(s_{ij} > s_{ik}|\mathbf{U}^g) \\ &= \prod_{(i,j,k)\in\Omega} \sigma(s_{ijk}) \end{aligned} \tag{5.5}$$

So, the goal is to find the latent representation $\mathbf{U}^g$ for network nodes which maximizes $P(> |\mathbf{U}^g)$ (i.e., to make preserving the aggregated ranking orders have maximum probability). It can be performed by minimizing the following sum of negative log-likelihood:

$$\begin{aligned}\min_{\mathbf{U}^g} \; L^g &= -\log P(> |\mathbf{U}^g) + \lambda_g ||\mathbf{U}^g||_F^2 \\ &= -\sum_{(i,j,k)\in\Omega} \log P(s_{ij} > s_{ik}|\mathbf{U}^g) + \lambda_g ||\mathbf{U}^g||_F^2 \\ &= -\sum_{(i,j,k)\in\Omega} \log \sigma(s_{ijk}) + \lambda_g ||\mathbf{U}^g||_F^2\end{aligned} \tag{5.6}$$

The connection between Equation 5.6 and Equation 5.3 is easy to see: log loss is used as the loss function $l(\cdot)$. Such a formulation provides a probabilistic interpretation for the ranking order preserving principle. Such a loss function is similar in spirit to the Bayesian Personalized Ranking (70), which attempts to predict the interaction between users and items.

5.4.2 Max Margin Representation Learning (MM-RL)

One can also employ a structural learning framework with max margin formulation as follows.

$$\begin{aligned}&\min_{\mathbf{U}^g} \sum_{(i,j,k)\in\Omega} \mu_{ijk} + \lambda_g ||\mathbf{U}^g||_F^2 \\ &\text{s.t. } s_{ijk} \geq 1 - \mu_{ijk}, \forall (i,j,k) \in \Omega\end{aligned} \tag{5.7}$$

where μ_{ijk} is a slack variable to impose soft margin. Such a formulation is similar to Structural SVM (39). To make clear its connection to the Equation 5.3 in the general framework, we can write it in the following form.

$$\min_{\mathbf{U}^g} \sum_{(i,j,k)\in\Omega} \max(0, 1 - s_{ijk}) + \lambda_g ||\mathbf{U}^g||_F \tag{5.8}$$

So, Equation 5.8 is equivalent to using hinge loss as $l(\cdot)$ in Equation 5.3.

The hinge loss is not differentiable at 0 and therefore poses difficulty for gradient-based optimization. We use a differentiable loss defined as follows.

$$l(x) = \begin{cases} 0 & \text{if } x \geq 2 \\ \frac{1}{4}(x-2)^2 & \text{if } 2 > x > 0 \\ 1 - x & \text{if } x \leq 0 \end{cases} \tag{5.9}$$

This loss function, which is also referred to as Huber loss, is a combination of L_1 loss (when $2 > x > 1$) and L_2 loss (when $x < 0$). In the link structure of many networks, there is often certain amount of noisy information. For example, it is not rare that a Facebook user may accept a connection invitation from someone he/she actually does not know (i.e., false positive), or two new users have not connected even if they know each other (i.e., false negative). Besides, the interaction between two proteins may have not been discovered due to the difficulty in the study of certain biological process (i.e., false negative). Such pairs might form noisy ranking triplets (i, j, k), which could potentially hamper the performance of link prediction models. Such noisy triplets might cause s_{ijk} to become negative. Rather than using

L_2 loss on the whole range, Huber loss uses L_1 loss for $x < 0$ because L_1 loss penalizes the error less harshly than L_2 loss and hence more robust to noisy triplets.

Hence, the optimization problem becomes the following:

$$\min_{\mathbf{U}^g} \; L^g = \sum_{(i,j,k)\in\Omega} l(s_{ijk}) + \lambda_g ||\mathbf{U}^g||_F^2 \tag{5.10}$$

where $l(\cdot)$ is the Huber loss defined in Equation 5.9.

5.5 Noise-resilient Representation Consensus Learning (NRCL)

We have discussed how to learn ranking-based representation from network links with P-RL and MM-RL. In this section, we describe the framework of NRCL based on learning representation consensus. Linkage information alone might not be sufficient for learning node representation, since network links are often sparse and noisy. Also, the node features can be of high dimensionality and contain many irrelevant features. Since links and attributes provide complementary information on the network nodes, it is desirable to learn a consensus from both the link-based and attribute-based representation. Also, the consensus learning enables link-only nodes and attribute-only nodes to be comparable in the latent space. Therefore, the similarity between the representations of two nodes can be used for cross view link prediction.

For the attribute-based representation, we learn a linear projection under the guidance of $\mathbf{U}^g$.

$$\min_{\mathbf{W}} \sum_{i\in O_s} ||\mathbf{U}_i^g - \mathbf{x}_i\mathbf{W}||_F^2 \tag{5.11}$$

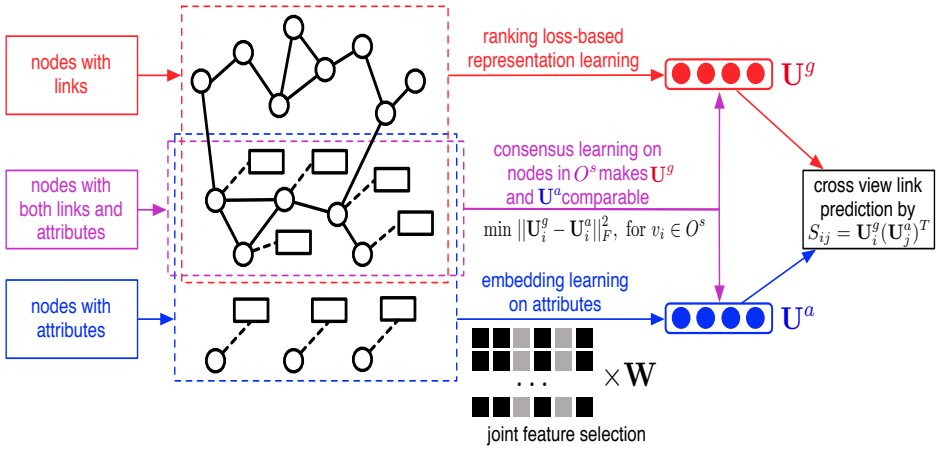


Figure 9: Representation consensus learning on partially observable networks

If we represent all the $\mathbf{U}_i^g$ and $\mathbf{x}_i$ in $i \in O_s$ as $\mathbf{U}^s$ and $\mathbf{X}^s$, respectively, we can write the objective function in the following form.

$$\min_{\mathbf{W}} ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 \tag{5.12}$$

Different features usually have different importances for predicting the links. For example, in the Facebook social network, "went to the same college" could be a more informative feature than "live in the same country" for link prediction. The projection matrix $\mathbf{W}$ can encode such knowledge by optimizing the objective in Equation 5.12 and useful features tend to have large (absolute value of) weights in the matrix $\mathbf{W}$.

Besides, node features could contain many irrelevant ones which could even harm the representation learning. To address this challenge, we propose to perform joint feature selection when learning the

projection. We use a feature selection indicator vector $\mathbf{s} \in \{0,1\}^D$ where $s_p = 1$ indicates the p-th feature is selected and $s_p = 0$ indicates the feature is not selected.

$$\begin{aligned} \min_{\mathbf{W},\mathbf{s}} \ & ||\mathbf{U}^s - \mathbf{X}^s \operatorname{diag}(\mathbf{s})\mathbf{W}||_F^2 \\ \text{s.t. } & s_p \in \{0,1\}, \forall p = 1,\ldots,D \\ & \sum_{p=1}^{D} s_p = d \end{aligned} \tag{5.13}$$

where $\operatorname{diag}(\mathbf{s})$ is the diagonal matrix with $\mathbf{s}$ as the diagonal elements. The constraint $\sum_{p=1}^{D} s_p = d$ means that we aim to select d ($d < D$) high quality features for the attribute-based representation. $\operatorname{diag}(\mathbf{s})\mathbf{W}$ is a matrix with d non-zero rows and hence it achieves feature selection. We combine $\mathbf{s}$ and $\mathbf{W}$ together, and employ $L_{2,0}$ norm to achieve the effect of feature selection:

$$\begin{aligned} \min_{\mathbf{W}} \ & ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 \\ \text{s.t. } & ||\mathbf{W}||_{2,0} \leq d \end{aligned} \tag{5.14}$$

The $L_{2,0}$ norm $||\mathbf{W}||_{2,0}$ is the number of rows in $\mathbf{W}$ with non-zero value. If $||\mathbf{W}_{i\cdot}||_F = 0$, i-th feature is not selected. We relax $||\mathbf{W}||_{2,0}$ to its convex hull, since the feasible region defined by $||\mathbf{W}||_{2,0} < d$ is not convex:

$$\begin{aligned} \min_{\mathbf{W}} \ & ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 \\ \text{s.t. } & ||\mathbf{W}||_{2,1} \leq d \end{aligned} \tag{5.15}$$

where the $L_{2,1}$ norm $||\mathbf{W}||_{2,1} = \sum_{i=1}^{D} ||\mathbf{W_i}.||_F$ could also achieve row sparsity. We further write the constraint in the form of Lagrangian as follows:

$$\min_{\mathbf{W}} L^a = ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 + \lambda_a||\mathbf{W}||_{2,1} \tag{5.16}$$

where λ_a is the regularization parameter on $L_{2,1}$ norm (63).

We combine the link-based loss and attribute-based loss together and the objective function becomes the following:

$$\begin{aligned}\min_{\mathbf{U}^g,\mathbf{W}} L &= L^g + L^a \\ &= \sum_{(i,j,k)\in\Omega} l(s_{ijk}) + \lambda_g||\mathbf{U}^g||_F^2 + \\ &\quad \alpha||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 + \lambda_a||\mathbf{W}||_{2,1}\end{aligned} \tag{5.17}$$

where α is the parameter that controls the relative importance of consensus learning. We refer to the instantiations of NRCL with L_g in Equation 5.6 and Equation 5.10 as P-NRCL and MM-RNCL, respectively.

Figure 9 summarizes the NRCL framework: 1) Representation $\mathbf{U}^g$ learned on linkage information might not be sufficiently good, as network links could be sparse and noisy. The consensus constraint $||\mathbf{U}_i^g - \mathbf{U}_i^a||_F^2$ (where the attribute-based representation $\mathbf{U}_i^a = \mathbf{x}_i\mathbf{W}$) serves as additional regularization on $\mathbf{U}^g$, which enables link-based representation to incorporate information from node attributes. This can be especially useful when a node has very few or no links. 2) On the other hand, node attributes are not equally important for link prediction. The consensus constraint $||\mathbf{U}_i^g - \mathbf{U}_i^a||_F^2$ can guide the

Algorithm 4 Alternating Optimization for NRCL

Initialize: $\mathbf{U}_i^g = rand(0,1)$, $\mathbf{W} = \mathbf{0}^{D\times m}$, $t = 1$.
while not converged **do**
  Fixing $\mathbf{W}$, find the optimal $\mathbf{U}^g$ by L-BFGS with Equation 5.25
  Fixing $\mathbf{U}^g$, find the optimal $\mathbf{W}$ with Algorithm 5
  $t = t + 1$
end while
$\mathbf{U}_i^a = \mathbf{x}_i\mathbf{W}, \forall i \in O^a$

learning of attribute-based representation by jointly selecting node attributes. By learning the consensus between $\mathbf{U}^g$ and $\mathbf{U}^a$, link information and attribute information could lend strength to each other for learning more noise-resilient representation. Also, the representations learned from network structure and node attributes become comparable in the latent space. To perform cross view link prediction, we can calculate the similarity $s_{ij} = \mathbf{U}_i^g(\mathbf{U}_j^a)^T$ in the latent space for link-only node v_i and attribute-only node v_j.

5.6 Optimization

In this section, we discuss how to solve the optimization problem for P-NRCL and MM-NRCL.

5.6.1 Alternating Optimization

For both instantiations, we need to optimize over $\mathbf{U}^g$ and $\mathbf{W}$. We decompose it to two sub-problems and develop an alternating optimization approach to solve the problem.

5.6.1.1 Fixing W, update $\mathbf{U}^g$

Now we derive the gradient for optimizing the objective function in Equation 5.6 and Equation 5.10. For P-NRCL, the gradient for one triplet is calculated as follows:

$$\frac{\partial l(s_{ijk})}{\partial \mathbf{U}_i^g} = \frac{e^{-s_{ijk}}}{1+e^{-s_{ijk}}} \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} \tag{5.18}$$

For Max Margin NRCL (MM-NRCL), the gradient is the following:

$$\frac{\partial l(s_{ijk})}{\partial \mathbf{U}_i^g} = \begin{cases} 0 & \text{if } s_{ijk} \geq 2 \\ \frac{1}{2}(s_{ijk}-2) \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} & \text{if } 2 > s_{ijk} > 0 \\ -\frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} & \text{if } s_{ijk} \leq 0 \end{cases} \tag{5.19}$$

The gradients on s_{ijk} w.r.t. $\mathbf{U}_i^g$, $\mathbf{U}_j^g$ and $\mathbf{U}_k^g$ are the following:

$$\frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} = \mathbf{U}_j^g - \mathbf{U}_k^g \tag{5.20}$$

$$\frac{\partial}{\partial \mathbf{U}_j^g} s_{ijk} = \mathbf{U}_i^g \tag{5.21}$$

$$\frac{\partial}{\partial \mathbf{U}_k^g} s_{ijk} = -\mathbf{U}_i^g \tag{5.22}$$

So, the gradient on L^g w.r.t $\mathbf{U}_i^g$ is as follows:

$$\begin{aligned}\frac{\partial L^g}{\partial \mathbf{U}_i^g} = &\sum_{(i,j,k)\in\Omega} \frac{e^{-s_{ijk}}}{1+e^{-s_{ijk}}} \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{ijk} + \\ &\sum_{(j,i,k)\in\Omega} \frac{e^{-s_{jik}}}{1+e^{-s_{jik}}} \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{jik} + \\ &\sum_{(j,k,i)\in\Omega} \frac{e^{-s_{jki}}}{1+e^{-s_{jki}}} \cdot \frac{\partial}{\partial \mathbf{U}_i^g} s_{jki}\end{aligned} \tag{5.23}$$

We can also derive the following gradient on L^a w.r.t $\mathbf{U}_i^g$:

$$\frac{\partial L^a}{\partial \mathbf{U}_i^g} = 2\alpha(\mathbf{U}_i^g - \mathbf{U}_i^a) \tag{5.24}$$

where $\mathbf{U}_i^a = \mathbf{x}_i\mathbf{W}$. To sum up, the gradient of the objective function in Equation 5.17 w.r.t $\mathbf{U}_i^g$ is as follows:

$$\frac{\partial L}{\partial \mathbf{U}_i^g} = \begin{cases} \frac{\partial L^g}{\partial \mathbf{U}_i^g} & \text{for } \phi(i) \in O^g \backslash O^s \\ \frac{\partial L^g}{\partial \mathbf{U}_i^g} + \frac{\partial L^a}{\partial \mathbf{U}_i^g} & \text{for } \phi(i) \in O^s \end{cases} \tag{5.25}$$

We can use gradient-based method (e.g., L-BFGS) to solve this subproblem.

5.6.1.2 Fixing $\mathbf{U}^g$, update $\mathbf{W}$

With fixed $\mathbf{U}^g$, we find the optimal $\mathbf{W}$ for the following convex sub-problem.

$$\min_{\mathbf{W}} \mathcal{L} = ||\mathbf{U}^s - \mathbf{X}^s\mathbf{W}||_F^2 + \lambda_a'||\mathbf{W}||_{2,1} \tag{5.26}$$

where $\lambda'_a = \lambda_a/\alpha$. To solve this subspace learning with $L_{2,1}$ regularization, we develop Algorithm 5 inspired by the iterative approach used in (63).

By setting $\frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{W}} = 0$, we have the following:

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{W}} = (\mathbf{X}^s)^T(\mathbf{X}^s\mathbf{W} - \mathbf{U}^s) + \lambda'_a\mathbf{E}\mathbf{W} = 0 \Rightarrow \\ \mathbf{W} = ((\mathbf{X}^s)^T\mathbf{X}^s + \lambda'_a\mathbf{E})^{-1}(\mathbf{X}^s)^T\mathbf{U}^s \end{aligned} \tag{5.27}$$

where $\mathbf{E}$ is a diagonal matrix with diagonal elements $\mathbf{E}_{ii} = \frac{1}{2||\mathbf{W}_i||_F}$ and $\mathbf{W}_i$ is the i-th row of $\mathbf{W}$.

Theorem 5.6.1 *For the optimization problem in Equation 5.26, Algorithm 5 would converge.*

Proof: It is easy to see that Equation 5.27 is a solution of the problem:

$$\min_{\mathbf{W}} \; ||\mathbf{X}^s\mathbf{W} - \mathbf{U}^s||_F^2 + \lambda'_a \operatorname{Tr}(\mathbf{W}^\mathbf{T}\mathbf{E}\mathbf{W}) \tag{5.28}$$

where $\operatorname{Tr}(\cdot)$ is the trace of matrix $(\cdot)$. So, from the t-th to $(t+1)$-th iteration,

$$\begin{aligned} &||\mathbf{X}^s\mathbf{W}^{t+1} - \mathbf{U}^s||_F^2 + \lambda'_a \operatorname{Tr}((\mathbf{W}^{t+1})^T\mathbf{E}^{t+1}\mathbf{W}^{t+1}) \\ &\leq ||\mathbf{X}^s\mathbf{W}^{t} - \mathbf{U}^s||_F^2 + \lambda'_a \operatorname{Tr}((\mathbf{W}^{t})^T\mathbf{E}^{t}\mathbf{W}^{t}) \Rightarrow \\ &||\mathbf{X}^s\mathbf{W}^{t+1} - \mathbf{U}^s||_F^2 + \lambda'_a \sum_i \frac{||\mathbf{W}_i^{t+1}||_F^2}{2||\mathbf{W}_i^t||_F} \\ &\leq ||\mathbf{X}^s\mathbf{W}^{t} - \mathbf{U}^s||_F^2 + \lambda'_a \sum_i \frac{||\mathbf{W}_i^{t}||_F^2}{2||\mathbf{W}_i^t||_F} \end{aligned} \tag{5.29}$$

Equivalently,

$$\begin{aligned} &||\mathbf{X}^s\mathbf{W}^{t+1} - \mathbf{U}^s||_F^2 + \lambda'_a||\mathbf{W}^{t+1}||_{2,1} - \\ &\lambda'_a(||\mathbf{W}^{t+1}||_{2,1} - \sum_i \frac{||\mathbf{W}_i^{t+1}||_F^2}{2||\mathbf{W}_i^t||_F}) \\ \leq\ &||\mathbf{X}^s\mathbf{W}^t - \mathbf{U}^s||_F^2 + \lambda'_a||\mathbf{W}^t||_{2,1} - \\ &\lambda'_a(||\mathbf{W}^t||_{2,1} - \sum_i \frac{||\mathbf{W}_i^t||_F^2}{2||\mathbf{W}_i^t||_F}) \end{aligned} \tag{5.30}$$

Note that $||\mathbf{W}^{t+1}||_{2,1} - \sum_i \frac{||\mathbf{W}_i^{t+1}||_F^2}{2||\mathbf{W}_i^t||_F} \leq ||\mathbf{W}^t||_{2,1} - \sum_i \frac{||\mathbf{W}_i^t||_F^2}{2||\mathbf{W}_i^t||_F}$ (because $\sqrt{a} - \frac{a}{2\sqrt{b}} \leq \sqrt{b} - \frac{b}{2\sqrt{b}}$, $a, b > 0$). So,

$$\begin{aligned} \mathcal{L}(\mathbf{W}^{t+1}) &= ||\mathbf{X}^s\mathbf{W}^{t+1} - \mathbf{U}^s||_F^2 + \lambda'_a||\mathbf{W}^{t+1}||_{2,1} \\ &\leq ||\mathbf{X}^s\mathbf{W}^t - \mathbf{U}^s||_F^2 + \lambda'_a||\mathbf{W}||_{2,1} = \mathcal{L}(\mathbf{W}^t) \end{aligned} \tag{5.31}$$

The objective function $\mathcal{L}(\mathbf{W})$ decreases in each iteration and it is lower bounded, so the convergence of Algorithm 5 is guaranteed. In our experiments, we observe it converges usually in less than 10 iterations.

Algorithm 4 summarizes the optimization methods for NRCL. The following theorem shows the convergence of this algorithm.

Theorem 5.6.2 *The alternating optimization framework in Algorithm 4 would converge.*

Proof: The objective function for each subproblem decreases in each iteration. The objective function in Equation 5.17 is hence guaranteed to decrease and it is lower-bounded. So the alternating optimization algorithm 4 would converge.

Algorithm 5 Algorithm for Learning Projection with $L_{2,1}$ norm

1: **Input**: $\mathbf{X^s} \in \mathcal{R}^{n_s \times D}$, projection target $\mathbf{U^s} \in \mathcal{R}^{n_s \times m}$, λ'_a
2: **Initialize**: $\mathbf{E} = \mathbf{I}_D$
3: **while** not converged **do**
4: $\quad \mathbf{W} = ((\mathbf{X}^s)^T\mathbf{X} + \lambda'_a\mathbf{E})^{-1}(\mathbf{X}^s)^T\mathbf{U}^s$
5: $\quad \mathbf{E} = \begin{bmatrix} \frac{1}{2||\mathbf{W}_1||_F} & & \\ & \dots & \\ & & \frac{1}{2||\mathbf{W}_D||_F} \end{bmatrix}$
6: **end while**
7: **Output**: $\mathbf{W} \in \mathcal{R}^{D \times m}$

5.6.2 Sampling ranking triplets

One can derive $O(|E||V|)$ triplets from the network structure. Such large amount of triplets is computationally expensive to optimize on. Rather than using all the potential triplets, we only sample a portion of them as follows: for each link (v_i, v_j) in the network, we randomly sample n_k ($n_k \ll |V|$) negative pairs to form triplets with (v_i, v_j). Hence, a total of $|E|n_k$ triplets (i.e, $|\Omega| = |E|n_k$) is used in the optimization. In our preliminary experiments, we found $n_k = 2$ or $n_k = 3$ is usually sufficient to achieve decent performance, so we use $n_k = 2$ in our experiments.

5.7 Experiments

In this section, we perform cross-view link prediction on four real-world networks with partially observable links and node attributes.

5.7.1 Datasets

We use four publicly available social/information network datasets in our experiments:

TABLE IX: Statistics of datasets

Statistics	Blogcatalog	Facebook	Wiki	Pubmed
# of instances	3192	1045	3363	19717
Avg. degree	8.87	51.19	19.76	4.50
# of attributes	3221	576	4973	500

- Facebook Dataset[1]: The whole dataset consists of ten ego-networks of facebook users. We use the network with largest number of nodes, which has 1045 users, 576 user profile features (e.g., education, work and location) and 53498 links.
- BlogCatalog Dataset[2]: We extract users who have blog posts in the category of $\{$Music, Finance, Health, Computers, Entertainment$\}$. The friendship connection between blog users establishes the network links and the word occurrence in the blogs is used as user features.
- Wikipedia Dataset[3] (73): Wikipedia articles from 19 categories and the hyperlinks establish the network structure. The original hyperlinks are directed and we symmetrize the network to make it undirected.

---

[1] `https://snap.stanford.edu/data/egonets-Facebook.html`

[2] `http://dmml.asu.edu/users/xufei/datasets.html`

[3] `http://linqs.cs.umd.edu/projects//projects/lbc/index.html`

- PubMed Dataset [3] (73): It consists of 19717 scientific publications about diabetes from PubMed database, which are classified into one of three classes. The word occurrence in the paper is used as the node features.

The statistics of these datasets are shown in Table IX.

5.7.2 Experimental Setting

TABLE X: Link prediction with different observable rates

Dataset	$\|O_s\|/n$	Metric	**Recommend AO to LO**						**Recommend LO to AO**					
			PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
Facebook	0.5770	Precision@5 (%)	39.47	**44.87**	34.34	41.58	27.24	8.68	21.53	**23.57**	16.69	17.71	11.85	8.79
		Recall@5 (%)	16.33	**20.09**	13.62	19.94	7.92	4.92	10.62	**12.41**	7.82	8.53	5.25	4.15
			PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
	0.3703	Precision@5 (%)	38.45	**39.48**	32.96	32.10	30.47	11.33	20.24	**22.51**	16.76	17.33	14.09	8.91
		Recall@5 (%)	10.26	**10.86**	8.03	9.66	6.65	3.31	6.63	**8.35**	4.59	5.52	3.84	4.13
BlogCatalog	0.5081		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	**14.65**	13.74	0.90	2.19	7.23	0.65	**1.18**	1.12	0.11	0.28	0.06	0.45
		Recall@5 (%)	**33.27**	30.82	2.12	4.68	15.08	1.59	**2.86**	1.93	0.23	0.66	0.14	1.01
	0.2701		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	13.43	**13.51**	0.41	1.45	8.24	0.74	**0.66**	0.44	0.09	0.13	0.06	0.63
		Recall@5 (%)	24.61	**24.83**	0.79	1.97	15.58	1.38	0.58	**0.59**	0.10	0.10	0.01	1.11
Wiki	0.5332		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	21.77	**24.42**	13.05	16.86	10.18	3.58	11.49	**14.12**	6.59	10.33	2.81	2.94
		Recall@5 (%)	25.91	**29.28**	11.50	16.52	11.31	4.49	14.78	**18.46**	7.75	12.51	3.11	2.58
	0.3417		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	21.01	**24.31**	13.23	17.83	9.90	3.59	11.05	**13.61**	3.96	7.87	1.09	3.51
		Recall@5 (%)	20.42	**25.48**	10.48	16.13	9.98	3.02	11.20	**14.18**	3.42	6.72	0.89	3.22
PubMed	0.4521		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	3.18	**4.90**	0.89	1.50	0.02	0.98	0.69	1.03	0.17	0.28	0.05	**1.44**
		Recall@5 (%)	7.92	**12.60**	1.61	2.93	0.03	2.31	1.57	2.63	0.22	0.57	0.18	**3.43**
	0.1847		PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk	PNRCL	MMNRCL	PRL	MMRL	LINE	DeepWalk
		Precision@5 (%)	2.57	**4.34**	0.41	1.08	0.03	0.82	0.44	0.69	0.10	0.19	0.02	**1.05**
		Recall@5 (%)	4.77	**9.66**	0.55	1.30	0.06	1.94	0.85	1.38	0.19	0.28	0.03	**2.38**

5.7.2.1 Baselines

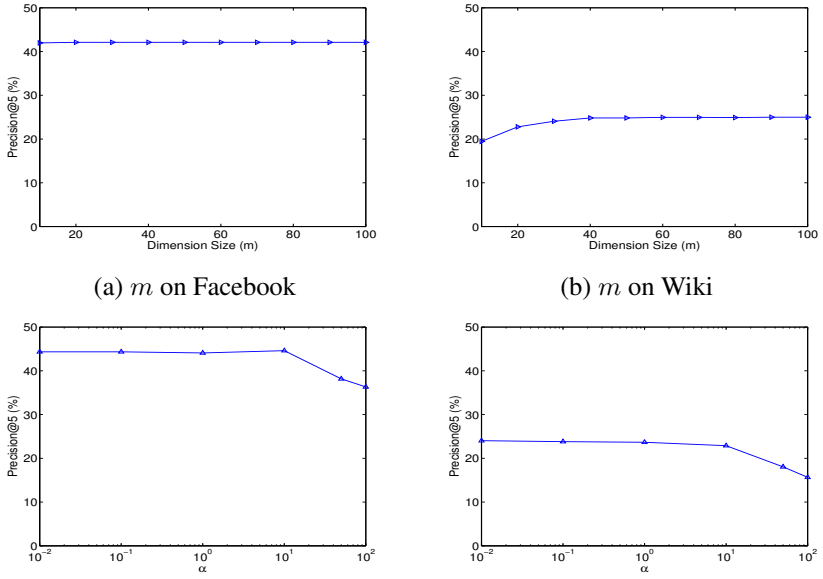
Existing methods usually assume the completeness of links and are not directly applicable for our problem setting. We create content links for each node in O^a (that has attributes) by connecting them with k other nodes with largest similarity w.r.t attributes, where k is the average degree of the network. Then we construct a combined network by connecting nodes that are connected by either a structural link or content link. We use the following methods on this combined network:

- Probabilistic Representation Learning (P-RL): P-RL learns the repsentation of nodes by optimizing the objective function L_g in Equation 5.6, which is similar to the triple loss based link prediction (56) (70).
- Max Margin Representation Learning(MM-RL): MM-RL learns the representation by optimizing the objective function L_g in Equation 5.10.
- LINE: An efficient embedding learning approach for network nodes (81) and the similarity between node embeddings can be used for link prediction.
- DeepWalk: It learns node representations that encode structural information by using truncated random walk as input (65). Recent work shows that it has state-of-the-art performance for link prediction (29).

5.7.2.2 Evaluation Metrics

We use the widely adopted metrics Precision, Recall to evaluate the performance of different link prediction approaches.

- $Precision@N = \frac{|C_{R_N} \cap C_{adopted}|}{N}$



(a) m on Facebook (b) m on Wiki

(c) α on Facebook (d) α on Wiki

Figure 10: Parameter sensitivity for MM-NRCL

- $Recall@N = \frac{|C_{R_N} \cap C_{adopted}|}{|C_{adopted}|}$

where C_{R_N} is the set of top N nodes in the recommendation and $C_{adopted}$ is the set of links that actually exist in the network. The precision and recall averaged over all the nodes are reported to reflect performance of each link prediction approaches.

5.7.2.3 Generating Partially Observable Networks

Similar to (94), we randomly select m_1 nodes and remove their links to create partially observable networks. We remove these links and denote the number of nodes without any links as m_2. Typically m_2 is larger than m_1, as removing the links for the m_1 nodes might make some other nodes become isolated. Then we pick another m_2 nodes randomly which have links and remove their attributes. Hence,

only $|O^s| = n - 2m_2$ nodes have both links and attributes. For recommending attribute-only nodes to link-only nodes (or link-only to attribute-only), 20% of the link-only (or attribute-only) nodes is used for validation and the rest is used for testing.

We set the dimension sizes (i.e., m) of P-NRCL, MM-NRCL, P-RL, MM-RL to 50 and that of LINE and DeepWalk to their default setting. For the regularization parameters in P-NRCL, MM-NRCL, P-RL and MM-RL, we perform grid search on the validation set in the following ranges: $\alpha = \{0.01, 0.1, 1, 10\}, \lambda_a = \{10, 20, 30\} \times \alpha, \lambda_g = \{5, 10, 15, 20\}$.

TABLE XI: Feature importance on Facebook dataset

Feature Name	**Feature Score**
Top ranked features	
education;school;id;anonymized feature 538	1.1095
education;school;id;anonymized feature 237	0.4649
work;employer;id;anonymized feature 151	0.4579
education;school;id;anonymized feature 52	0.3724
education;concentration;id;anonymized feature 14	0.3499
Examples of unselected features	
last_name;anonymized feature 592	0
work;employer;id;anonymized feature 648	0
work;position;id;anonymized feature 697	0
work;end_date;anonymized feature 674	0
education;school;id;anonymized feature 459	0

5.7.3 Comparison on Cross View Link Prediction

We report the link prediction performance with different percentages ($|O_s|/n$) of fully observable nodes in Table X by setting $m_1 = \{0.2, 0.3\} \times n$. On most of the datasets, NRCL methods (especially MM-NRCL) outperform the baseline methods significantly. For example, on Wiki dataset, P-NRCL and MM-NRCL improve over the best baseline method MM-RL by 29.1% and 44.8%, respectively, in terms of precision@5. When the fully observable rate goes to as low as $20\% \sim 40\%$, MM-NRCL still performs very well for cross view link prediction. Though MM-RL and P-RL employ the same objective function L_g on the link-based representation learning as MM-NRCL and P-NRCL, the content links created from potentially noisy feature space make them unable to learn high quality representation. This indicates the importance of selecting the most informative features for representation learning on partially observable networks, in order to achieve decent link prediction performance. In comparison, the representation learned by MM-NRCL and P-NRCL could be more resilient to irrelevant features, as NRCL performs joint feature selection and only use the high-quality features to learn the representation.

When comparing P-NRCL with MM-NRCL, we observe that MM-NRCL performs better than P-NRCL in most cases. Similarly, MM-RL often outperforms P-RL as well. This suggests that Huber loss, which is more robust to noisy links, tends to be a better choice for learning node representations than log loss.

5.7.4 Case Study on Joint Feature Selection

Since we perform joint feature selection in our NRCL framework, the utility of feature i ($i = 1, 2, \ldots, D$) can be ranked by their coefficients $\sum_{j=1}^{m} W_{ij}^2$. For useless features, $\sum_{j=1}^{m} W_{ij}^2$ tends to shrink towards zero under the effect of $L_{2,1}$ norm, while important features tend to have large values of

$\sum_{j=1}^{m} W_{ij}^2$. As a case study, we show the feature importance for Facebook dataset in Table XI. The specific value and meaning of features are anonymized for privacy concern. Features in the same category (e.g., education) can be encoded into multiple binary features and they may have different importance for representation learning. For instance, *education features* 538 and 237 are highly important while *education feature* 459 is considered useless. By examining the features with large scores ($\sum_{j=1}^{m} W_{ij}^2$), one could have a deeper understanding about the roles of different features in the formation of network links.

5.7.5 Sensitivity Analysis

In this subsection, we study the effect of dimension size m and consensus regularization parameter α only for MM-NRCL, since previous results show that MM-NRCL is more promising than P-NRCL. The precision results w.r.t different parameter values on Facebook and Wiki datasets are shown in Figure 10.

Effect of latent dimension size For m, we can observe that MM-NRCL is not very sensitive to the parameter value when it is not too small (e.g., $m \leq 20$).

Effect of regularization controlling consensus strength For the co-regularization parameter α, MM-NRCL can perform consistently well as long as α is not too large (e.g., $\alpha \geq 10$).

CHAPTER 6

MULTI-VIEW UNSUPERVISED FEATURE SELECTION BY CROSS-DIFFUSED MATRIX ALIGNMENT

(This chapter was accepted/to appear as "Multi-view Unsupervised Feature Selection by Cross-diffused Matrix Alignment", in Proceedings of the 30th International Joint Conference on Neural Networks (IJCNN 17), 2017)

6.1 Introduction

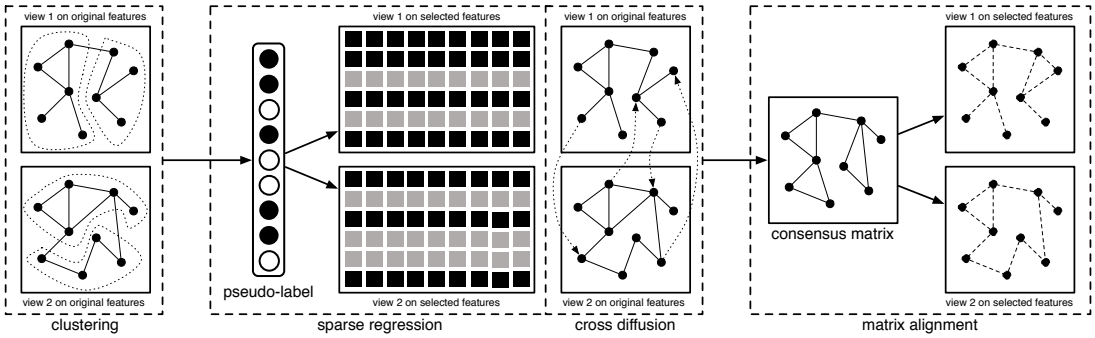
Data obtained from different sources or feature subsets usually provide complementary information for machine learning tasks, and conventionally they are named as multi-view data. We can observe multi-view data in a wide range of application domains. For example, news about the same event can often be reported in different languages and by different agencies. In the video domain, in addition to features extracted from visual signals, videos are often equipped with textual descriptions and related tags. In medical science, various diagnosis tools can provide many measurements from different laboratory tests, including clinical, imaging, immunologic and serologic features.

Capability for simultaneous consideration of data coming from multiple views/sources is important for many learning tasks, which is referred to as multi-view learning. Multiple views together depict an enriched picture about the entities of interest and thereby provide an effective way of heterogeneous data fusion. How to effectively incorporate the abundant information from multiple views is critical for different application domains (43) (82). It has been shown that incorporating information from multiples

views can improve the performance of various machine learning tasks. For example, co-regularized spectral clustering (43), by enforcing consensus learning on latent factors, outperforms single-view clustering significantly.

The curse of dimensionality is an inevitable problem in the era of big data, which is also one of the major challenges in many multi-view learning scenarios. For example, the vocabulary of news articles can contain more than $100,000$ words in each language. Also, the user generated content in social media (such as blog websites) tends to be highly noisy. Such high-dimensional noisy data can hamper the performance and efficiency of many machine learning/data mining tasks. Feature selection is potentially a useful technique for alleviating such issue. Traditional feature selection methods mainly focus on a single view which could be insufficient considering the existence of other views being available. It is desirable to utilize information from other complementary views, when selecting features for each view.

Since class labels are usually expensive to obtain, unsupervised feature selection usually has wider applicability than its supervised counterpart. The key challenge of unsupervised multi-view feature selection is twofold: (1) how to effectively represent the fused information from multiple views, and (2) how to effectively exploit the fused information representation to select high-quality features. State-of-the-art unsupervised multi-view feature selection approaches (82) (69) fuse information by generating intermediate cluster labels. However, summarizing the information for each instance with a cluster label tends to lose too much information, since the cluster labels are usually noisy and inaccurate. In this paper, we propose a new method, CDMA-FS (Cross Diffused Matrix Alignment based Feature Selection), to address the challenges of multi-view feature selection in unsupervised setting. The advantages of our method compared to state-of-the-art approaches (82) (69) can be summarized as follows.



(a) Pseudo-label based multi-view feature selection(b) Multi-view feature election by cross-diffused matrix alignment

Figure 11: Comparison of CDMA-FS Framework with existing multi-view feature selection methods

- We employ a cross diffusion-based approach to learn a consensus similarity graph from multiple views, which retains more information than the cluster labels (Figure 11).
- Rather than relying on cluster-label guided sparse regression, we directly exploit the information from the cross-diffused matrix by matrix alignment.
- Existing approaches typically have a few parameters which are difficult to set in unsupervised setting. This makes them less practical for real-world applications. In contrast, we provide guidelines for setting the parameter in the proposed method.
- Our objective function is not based on linear regression and hence can evaluate the non-linear usefulness of features.

6.2 Related Work

Earlier unsupervised feature selection methods (34) (109) usually assign scores to each feature based on certain heuristics and neglect the correlation among features. However, such heuristic based meth-

ods usually ignore the correlation among the features and redundancy may exist in the selected features. In recent years, different methods (106) (68) (99) (96) have been proposed to evaluate feature quality jointly. Linear projection based methods (106) (48) (21) (90) with sparsity-inducing $L_{2,1}$ norm have become prevalent among others. Compared to the heuristic-based methods (34) (109), the major advantage of $L_{2,1}$-based approaches is that they can evaluate features jointly. Different $L_{2,1}$ norm-based methods usually differ in the ways they generate pseudo labels and the loss functions on the projection. Unsupervised Discriminative Feature Selection (UDFS) (106) introduces pseudo-label based regression to better capture the information from the local structure. Non-negative Discriminative Feature Selection (NDFS) (48) derives the cluster/pseudo labels from non-negative spectral analysis. Robust Unsupervised Feature Selection (RUFS) (68) and Embedded Unsupervised Feature Selection (EUFS) (90) generate pseudo labels from non-negative matrix factorization. Robust Spectral Feature Selection (RSFS) (76) employs local kernel regression for the cluster indicators and Huber loss for the projection. These methods are only able to evaluate the. To address this issue, Stochastic Neighbor-preserving Feature Selection (SNFS) (101) and Nonlinear Joint Feature Selection (NJFS) (95) are proposed, which can evaluate the non-linear usefulness of features.

Recently, several pseudo label-based methods have been extended to multi-view setting (82) (69) (75) via cluster consensus learning. In these approaches, pseudo-labels derived from certain clustering algorithms are required to be the same across different views in order to incorporate multi-view information. For example, adaptive Unsupervised Multi-view Feature Selection (AUMFS) (25) rely on spectral clustering on the combined similarity graphs obtained from different views. Multi-View Feature Selection (MVFS) (82) and MVUFS (69) can be seen as extention of NDFS (48) and RUFS (68)

to multi-view feature selection by enforcing consensus on the cluster indicators from different views, respectively. However, they rely on the cluster labels to guide feature selection, and the noisy cluster labels may lead to suboptimal feature selection results. Also, they evaluate features based on linear regression and hence cannot select high-quality features if they are non-linearly correlated with the class labels.

6.3 Fusing Different Views by Cross Diffusion

We denote n data samples with m views as $\{\mathbf{X}^{(v)}|v = 1, \ldots, m\}$, $\mathbf{X}^{(v)} = [\mathbf{x}_1^{(v)}, \mathbf{x}_2^{(v)}, \ldots, \mathbf{x}_n^{(v)}]$ and the number of features in the v-th view as $D^{(v)}$. So $\mathbf{x}_i^{(v)} \in \mathbb{R}^{D^{(v)}}$ and $x_{ip}^{(v)}$ denotes the value of p-th ($p = 1, \ldots, D^{(v)}$) feature of $\mathbf{x}_i^{(v)}$.

The proposed CDMA-FS framework is a two-step approach. First, we fuse different kernels into one robust similarity matrix through cross diffusion. Second, we perform matrix alignment for the features from each view so that the kernel constructed from the selected features can best align with the fused matrix (Figure 11). In this manner, feature selection on each view can benefit from the consensus information fused from multiple views.

With the features from the v-th view, one can construct a kernel/similarity matrix for this view. There are different types of similarity matrices:

- Gaussian Kernel Weighting: $W_{ij} = e^{-(\mathbf{x}_i - \mathbf{x}_j)^2/\sigma^2}$
- Dot-product Kernel Weighting: $W_{ij} = \mathbf{x}_i^T \cdot \mathbf{x}_j$
- 0-1 Weighting: $W_{ij} = 1$ if and only if $\mathbf{x}_i$ is within $\mathbf{x}_j$'s k Nearest Neighbors.

A similarity matrix can then be used to define the transition probability as follows.

$$\mathcal{P}_{ij}^{(v)} = \frac{W_{ij}^{(v)}}{\sum_{k=1}^{n} W_{ik}^{(v)}} \tag{6.1}$$

where $\sum_{j=1}^{n} \mathcal{P}_{ij}^{(v)} = 1$ ($\forall i = 1, \ldots, n$) and we let $\mathcal{P}_{ii}^{(v)} = 0$ for convenience. For a probability vector $\mathbf{u}$ (i.e., $\mathbf{u}^T\mathbf{1} = 1$), $\mathbf{u}^T\boldsymbol{\mathcal{P}}^{(v)}$ is a Markov random walk of $\mathbf{u}$ w.r.t. $\boldsymbol{\mathcal{P}}^{(v)}$. $\boldsymbol{\mathcal{P}}^{(v)}\mathbf{u}$ can be viewed as a local averaging operation with $\mathbf{W}^{(v)}$ measuring the locality. It can also be interpreted as a generalization of Parzen window estimators to functions on the local manifold (88). Both $\mathbf{u}^T\boldsymbol{\mathcal{P}}^{(v)}$ and $\boldsymbol{\mathcal{P}}^{(v)}\mathbf{u}$ can be viewed as a diffusion process.

6.3.1 Cross Diffusion

Cross diffusion (88) aims to exploit mutual enhancement of different views inspired by co-training (8). The main idea of cross diffusion is to perform random walk using the transition probability from different views in an alternating manner. In the case of $m = 2$, the cross diffusion process can be defined as follows.

$$\mathbf{P}_{t+1}^{(1)} = \boldsymbol{\mathcal{P}}^{(1)} \cdot \mathbf{P}_{t}^{(2)} \cdot (\boldsymbol{\mathcal{P}}^{(1)})^T \tag{6.2}$$

$$\mathbf{P}_{t+1}^{(2)} = \boldsymbol{\mathcal{P}}^{(2)} \cdot \mathbf{P}_{t}^{(1)} \cdot (\boldsymbol{\mathcal{P}}^{(2)})^T \tag{6.3}$$

where $\mathbf{P}_t^{(1)}$ and $\mathbf{P}_t^{(2)}$ are the status matrices at the t-th iteration for view 1 and view 2, respectively. For the initial values, we set $\mathbf{P}_1^{(1)} = \boldsymbol{\mathcal{P}}^{(1)}$ and $\mathbf{P}_1^{(2)} = \boldsymbol{\mathcal{P}}^{(2)}$. Since the distances between data points are usually unreliable in high-dimensional space, it is usually preferable to use the k nearest neighbors as $\boldsymbol{\mathcal{P}}^{(1)}$ and $\boldsymbol{\mathcal{P}}^{(2)}$. Under mild conditions that $\boldsymbol{\mathcal{P}}^{(1)}$ and $\boldsymbol{\mathcal{P}}^{(2)}$ are irreducible and aperiodic, the convergence

of this process can be proved using Perron-Frobenius Theorem (66). The final status matrix can be computed as the average of status matrices from two views: $\mathbf{P}^* = (\mathbf{P}_e^{(1)} + \mathbf{P}_e^{(2)})/2$, where e is the number of iterations at which the cross diffusion terminates. We refer to this final status matrix $\mathbf{P}^*$ as *cross diffused matrix*.

Let us denote the connected components in the cross-diffused matrix as $\{\theta_1, \theta_2, \ldots, \theta_Q\}$, where Q is the total number of connected components. We also denote the ground-truth class label of $\mathbf{x}$ as $c(\mathbf{x})$. We define the purity of the q-th connected component as the percentage of majority class of instances. If $purity(\theta_q) \geq 1 - \epsilon$ for all $1 \leq q \leq Q$, we say that $\mathbf{P}$ is an ϵ**-good graph**. At the $(2t+1)$-th iteration, $\mathbf{P}_{2t+1}^{(1)}$ and $\mathbf{P}_{2t+1}^{(2)}$ can be written as the following.

$$\mathbf{P}_{2t+1}^{(1)} \propto (\boldsymbol{\mathcal{P}}^{(1)}\boldsymbol{\mathcal{P}}^{(2)})^t \cdot \boldsymbol{\mathcal{P}}^{(2)} \cdot ((\boldsymbol{\mathcal{P}}^{(2)})^T(\boldsymbol{\mathcal{P}}^{(1)})^T)^t \tag{6.4}$$

$$\mathbf{P}_{2t+1}^{(2)} \propto (\boldsymbol{\mathcal{P}}^{(2)}\boldsymbol{\mathcal{P}}^{(1)})^t \cdot \boldsymbol{\mathcal{P}}^{(1)} \cdot ((\boldsymbol{\mathcal{P}}^{(1)})^T(\boldsymbol{\mathcal{P}}^{(2)})^T)^t \tag{6.5}$$

In order to effectively guide subsequent feature selection, it is desirable that the connected components in $\mathbf{P}_{2t+1}^{(1)}$ and $\mathbf{P}_{2t+1}^{(2)}$ obtained from the cross-diffusion process have large purity. The following theorem provides guarantee on the purity of components in the cross-diffused matrix (88).

Theorem 6.3.1 *If the K-nearest-neighbors is good to measure local affinity (93),* $\mathbf{P}_{2t+1}^{(1)}$ *and* $\mathbf{P}_{2t+1}^{(2)}$ *are* ϵ*-good graphs. The number of connected components in graph* $\mathbf{P}_{2t+1}^{(1)}$ *is equal to that of graph* $\mathbf{P}_{2t+1}^{(2)}$*, and it is not larger than that in graphs* $\boldsymbol{\mathcal{P}}^{(1)}$ *and* $\boldsymbol{\mathcal{P}}^{(2)}$*.*

Moreover, it is usually helpful to add regularization at each iteration of the diffusion process to make the probability matrix more robust.

$$\mathbf{P}_{t+1}^{(1)} = \boldsymbol{\mathcal{P}}^{(1)} \cdot \mathbf{P}_{t}^{(2)} \cdot (\boldsymbol{\mathcal{P}}^{(1)})^T + \alpha \mathbf{I} \tag{6.6}$$

$$\mathbf{P}_{t+1}^{(2)} = \boldsymbol{\mathcal{P}}^{(2)} \cdot \mathbf{P}_{t}^{(1)} \cdot (\boldsymbol{\mathcal{P}}^{(2)})^T + \alpha \mathbf{I} \tag{6.7}$$

where $\mathbf{I}$ is an identity matrix and α is the parameter that controls the regularization. We remark that CDMA-FS can perform reasonably well for a wide range of α (e.g., $10^{-4} \sim 10$).

6.3.2 Extension to more than two views

Similar to the case of $m = 2$, $\mathbf{P}_{t+1}^{(v)}$ for $m > 2$ can be calculated as follows.

$$\mathbf{P}_{t+1}^{(v)} = \boldsymbol{\mathcal{P}}^{(v)} \cdot \frac{1}{m-1} \sum_{i \neq v} \mathbf{P}_{t}^{(i)} \cdot (\boldsymbol{\mathcal{P}}^{(v)})^T \tag{6.8}$$

The final status matrix is the average of m matrices:

$$\mathbf{P}^* = \frac{1}{m} \sum_{v=1}^{m} \mathbf{P}_{e}^{(v)} \tag{6.9}$$

Since the transition probability might be not reliable for non-nearest neighbors, we create a kNN graph $\mathbf{G}$ from $\mathbf{P}^*$ after obtaining $\mathbf{P}^*$. In the following section, we present how to use $\mathbf{G}$ to guide the feature selection for each view.

6.4 Aligning with Cross-diffused Matrix

Our goal is to select $d^{(v)}$ ($d^{(v)} \ll D^{(v)}$) high-quality features for each view. We denote the selection indicator vector as $\mathbf{s}^{(v)} \in \{0,1\}^{D^{(v)}}$, where $s_p^{(v)} = 1$ indicates that the p-th feature is selected and $s_p^{(v)} = 0$ otherwise.

To directly exploit the information from the cross-diffused matrix for feature selection in each view, we propose to perform matrix alignment towards the cross-diffused matrix. We assume that a kernel matrix can be constructed from each view based on the selected features $\mathrm{diag}(\mathbf{s})\mathbf{X}^{(v)}$ with Gaussian kernels (i.e., Radial Basis Function):

$$K_{ij}^{(v)} = \exp\left(-\frac{1}{\sigma^2}\|\mathrm{diag}(\mathbf{s}^{(v)})\mathbf{x}_i^{(v)} - \mathrm{diag}(\mathbf{s}^{(v)})\mathbf{x}_j^{(v)}\|^2\right) \tag{6.10}$$

The intuitive idea of CDMA-FS is to make the kernel constructed from selected features imitate the cross-diffused matrix $\mathbf{G}$. We achieve this by employing the matrix alignment technique (20) (94) as follows.

Definition 9 Matrix Alignment *For two symmetric matrices $\mathbf{K}_1 \in \mathbb{R}^{n\times n}$ and $\mathbf{K}_2 \in \mathbb{R}^{n\times n}$, the alignment between $\mathbf{K}_1$ and $\mathbf{K}_2$ is defined as*

$$\rho(\mathbf{K}_1, \mathbf{K}_2) = \frac{\mathrm{Tr}(\mathbf{K}_1\mathbf{K}_2)}{||\mathbf{K}_1||_F \cdot ||\mathbf{K}_2||_F} \tag{6.11}$$

where $\mathrm{Tr}(\cdot)$ is the trace of a matrix.

Matrix alignment can be viewed as computing the cosine similarity between two vectorized matrices. However, the optimization problem is more difficult to solve with the normalization term $||\mathbf{K}_1||_F \cdot ||\mathbf{K}_2||_F$. In this paper, we employ the unnormalized version of matrix alignment as in (20), which can be considered as the inner product between two vectorized matrices.

Definition 10 Unnormalized Matrix Alignment *For two symmetric matrices* $\mathbf{K}_1 \in \mathbb{R}^{n\times n}$ *and* $\mathbf{K}_2 \in \mathbb{R}^{n\times n}$*, the alignment between* $\mathbf{K}_1$ *and* $\mathbf{K}_2$ *is defined as*

$$\rho(\mathbf{K}_1, \mathbf{K}_2) = \mathrm{Tr}(\mathbf{K}_1\mathbf{K}_2) \tag{6.12}$$

It is usually helpful to center the matrix for better matrix alignment performance as in observed in (19). For a symmetric matrix $\mathbf{K}$, centering $\mathbf{K}$ can be achieved by $\mathbf{HKH}$, where the centering matrix $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$.

Definition 11 Centered Matrix Alignment *For two real matrices* $\mathbf{K}_1 \in \mathbb{R}^{n\times n}$ *and* $\mathbf{K}_2 \in \mathbb{R}^{n\times n}$*, the centered alignment between* $\mathbf{K}_1$ *and* $\mathbf{K}_2$ *is defined as*

$$\begin{aligned}\rho(\mathbf{K}_1, \mathbf{K}_2) &= \mathrm{Tr}(\mathbf{H}\mathbf{K}_1\mathbf{H}\mathbf{H}\mathbf{K}_2\mathbf{H}) \\ &= \mathrm{Tr}(\mathbf{H}\mathbf{K}_1\mathbf{H}\mathbf{K}_2)\end{aligned}$$

where the second equation can be obtained by noting $\mathbf{HH} = \mathbf{H}$ *and* $\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA})$ *for arbitrary matrices* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$.

After a high-quality cross-diffused matrix is obtained, we select features for each view under the guidance of this matrix. To achieve this, we aim to maximize the correlation between the cross-diffused matrix and the kernel matrix computed from selected features. To select $d^{(v)}$ features for the v-th view, we formulate it as a constrained optimization problem and find $\mathbf{s}^{(v)}$ to minimize the following objective function:

$$\begin{aligned} \min_{\mathbf{s}^{(v)}} \ & f = -\operatorname{Tr}(\mathbf{HGHK}^{(v)}) \\ \text{s.t.} \ & \sum_{p=1}^{D^{(v)}} s_p^{(v)} = d^{(v)} \\ & s_p^{(v)} \in \{0,1\}, \forall p = 1, \ldots, D^{(v)} \end{aligned} \tag{6.13}$$

Discussion Traditional sparse regression based methods (82) (69) rely on generating intermediate cluster labels and rank features by their linear regression coefficients. In contrast, CDMA-FS framework utilizes the cross-diffused matrix, which preserves more information than cluster labels. Also, the connected components in the cross diffused matrix tend to have good purity as shown in Theorem 1, which means the connected data points are likely from the same class. The objective, through matrix alignment, aims to select the features that make connected instances close and unconnected instances far apart. By optimizing the objective above, we directly infer the selection vector $\mathbf{s}$ which can achieve the following desirable effects: features that make data points from the same class similar would be rewarded and features that make data points from different classes similar would shrink s_p to zero. Hence, different classes would be more separable in the space of selected features.

6.5 Optimization

6.5.1 Gradient Derivation with Relaxed Constraint

The '0/1' integer programming problem in Equation 6.13 is computationally intensive to optimize. We relax the '0/1' constraint on $s_p^{(v)}$ ($p = 1, \ldots, D^{(v)}$) to real values in range of $[0, 1]$ to make the optimization tractable as in (95). We further rewrite the summation constraint $\sum_{p=1}^{D^{(v)}} s_p^{(v)} = d^{(v)}$ in the form of Lagrange multiplier:

$$\begin{aligned} \min_{\mathbf{s}^{(v)}} \; & f = -\operatorname{Tr}(\mathbf{HGHK}^{(v)}) + \lambda ||\mathbf{s}^{(v)}||_1 \\ & \text{s.t. } 0 \leq s_p^{(v)} \leq 1, \forall p = 1, \ldots, D^{(v)} \end{aligned} \tag{6.14}$$

where $||\cdot||_1$ denotes the l_1 norm on vector $(\cdot)$ and λ controls the sparsity of $\mathbf{s}^{(v)}$. Note that in our case $||\mathbf{s}^{(v)}||_1 = \sum_{p=1}^{D} s_p$ since we have non-negative constraints on $\mathbf{s}^{(v)}$.

We can derive the following gradient w.r.t. the objective function, since $\mathbf{K}^{(v)}$ ($v = 1, \ldots, m$) is a symmetric matrix.

$$\begin{aligned} \frac{\partial f}{\partial s_p^{(v)}} &= -\sum_{i,j=1}^{n} ((\mathbf{HGH})_{ij} \cdot \frac{\partial K_{ij}^{(v)}}{\partial s_p^{(v)}}) + \lambda \\ &= \sum_{i,j=1}^{n} (((\mathbf{HGH}) \odot \mathbf{K}^{(v)})_{ij} \left(x_{ip}^{(v)} - x_{jp}^{(v)} \right)^2) \frac{2 s_p}{\sigma^2} + \lambda \end{aligned} \tag{6.15}$$

where $\odot$ is element-wise product. To solve this constrained optimization problem efficiently, we use Projected Quasi-Newton Method as shown in the next subsection.

6.5.2 Projected Quasi-Newton Method

Traditional Newton method optimizes the following second-order approximation at the t-th iteration.

$$q_t(\mathbf{s}) = f(\mathbf{s}_t) + (\mathbf{s} - \mathbf{s}_t)^T \nabla f(\mathbf{s}_t) + \frac{1}{2}(\mathbf{s} - \mathbf{s}_t)^T \mathbf{B}_t (\mathbf{s} - \mathbf{s}_t) \tag{6.16}$$

where $\mathbf{B}_t = \nabla^2 f(\mathbf{s}_t)$ is the Hessian matrix. Newton method enjoys good convergence rate but the Hessian matrix requires $O(D^2)$ storage and it is time-consuming to compute. So Quasi-Newton methods (e.g., L-BFGS (52)) use a positive definite approximation to the Hessian matrix $\nabla^2 f(\mathbf{s}_t)$. For example, L-BFGS (52) uses the gradients in previous iterations to compute an approximate Hessian matrix.

$$\mathbf{B}_{t+1} = \mathbf{B}_t - \frac{\mathbf{B}_t \mathbf{u}_t \mathbf{u}_t^T \mathbf{B}_t}{\mathbf{u}_t^T \mathbf{B}_t \mathbf{u}_t} + \frac{\mathbf{y}_t \mathbf{y}_t^T}{\mathbf{y}_t^T \mathbf{u}_t} \tag{6.17}$$

where $\mathbf{u}_t = \mathbf{s}_{t+1} - \mathbf{s}_t$ and $\mathbf{y}_t = \nabla f(\mathbf{s}_{t+1}) - \nabla f(\mathbf{s}_t)$.

To address the constraints on $\mathbf{s}$ in Equation 6.14, projected Newton method can be used to solve the following constrained quadratic approximation:

$$\begin{aligned} \min_{\mathbf{s}} \quad & q_t(\mathbf{s}) \\ \text{s.t.} \quad & \mathbf{s} \in \mathcal{C} \end{aligned} \tag{6.18}$$

In our case, $\mathcal{C}$ is the $[0, 1]$ box constraint on $\mathbf{s}^{(v)}$. A projection operator for this constraint can be defined as follows.

$$[\text{Proj}_{[0,1]}(\mathbf{s}^{(v)})]_p = \min(1, \max(0, s_p^{(v)})), \ \forall p = 1, 2, \ldots, D^{(v)} \tag{6.19}$$

To make the optimization more efficient, we use a variant of the L-BFGS method which employs spectral projected gradient method as subroutine to solve the constrained problem in Equation 6.18. The optimization method (71) is two-level approach: at the outer level, L-BFGS updates are used to construct a sequence of quadratic approximations (with constraints) to the problem; at the inner level, a spectral projected gradient method optimizes the constrained subproblem approximately to generate a feasible direction. The number of iterations in this algorithm remains linear in dimensionality of feature vector, but with a higher constant factor than the L-BFGS method. Nevertheless, the method can lead to significant gain when the cost of the projection is much lower than evaluating the function, which is the case in our problem setting.

Although we could use spectral projected gradient method to exactly solve problem Equation 6.18, it is expensive to do so in practice. Therefore, we terminate the spectral gradient descent subroutine before the exact solution is found, since our goal is only to obtain a feasible descent direction for L-BFGS. One might be concerned about the early termination of the spectral gradient descent subroutine, but in (71) it has been shown that the spectral gradient descent subroutine, even when terminated early, can give a descent direction, if we initialize it with $\mathbf{s}_t$ and we perform at least one spectral gradient descent iteration. In the implementation, we can parametrize the maximum number of the spectral gradient descent iterations by t_p, the cost of one iteration is $O(mt_pD)$ for the inexact Newton method, given that our projection operation requires $O(D)$ time and L-BFGS stores m most recent gradients. The projected Quasi-Newton algorithm is shown in Algorithm 6.

Algorithm 6 Solve CDMA-FS with Projected Quasi-Newton Algorithm

Initialize: $\mathbf{s}_0 \leftarrow \mathbf{1}$, $t = 0$.
while not converged **do**
  Compute the gradient by Equation 6.15
  Compute the approximate Hessian
  Solve Equation 6.18 for $\mathbf{s}_t^*$ using projected spectral gradient algorithm.
  $\mathbf{d}_t = \mathbf{s}_t^* - \mathbf{s}_t$
  Perform line search on the direction of $\mathbf{d}_t$ to satisfy the Armijo condition.
  $t = t + 1$
end while
Select the features with corresponding entry in $\mathbf{s}$ equal to 1.

6.6 Parameter Selection

Existing multi-view feature selection methods typically have $2 \sim 3$ regularization parameters and it is difficult to choose appropriate values for these parameters when class labels are not available. In the original papers of these psuedo-label approaches (106) (68) (76), only the best performance is reported, the parameters of which are tuned using all the class labels. However, such way of setting parameters violates the assumption of no supervision. In practice, it is impossible to know the best parameter values and this makes them less useful for real world applications.

For CDMA-FS, we provide guidelines for choosing the value of parameter λ. Let us denote the number of features with $s_p^{(v)} = 1$ as $N_1^{(v)}$, which is influenced by the value of λ. By noting that $N_1^{(v)}$ is a monotonically non-increasing function of λ, we can choose the value of λ for each view that makes $N_1^{(v)}$ equal to (or within a small range of) the feature size one wants to retain.

TABLE XII: Statistics of datasets

Statistics	Reuters		BBC Sport		BlogCatalog		CNN	
# of instances	1575		544		1000		2107	
# of features	view1 3791	view2 2862	view1 3183	view2 3203	view1 5390	view2 2003	view1 6262	view2 996
# of classes	6		5		5		7	

6.7 Experiments

In this section, we compare the proposed method with state-of-the-art baseline methods on four real world datasets.

6.7.1 Datasets

We use four publicly available real-world datasets in our experiments.

- Reuters Multilingual dataset [1]: News articles in English and German on six topics. Each language can be considered a view for the same article.
- BBC Sport dataset [2]: BBC news articles from 5 topics: *athletics, cricket, football, rugby, tennis*. Paragraphs in the news articles are used to construct two views.

---

[1] `https://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2+Multilingual,+Multiview+Text+Categorization+Test+collection`

[2] `http://mlg.ucd.ie/datasets/segment.html`

- CNN dataset [1]: It consists of news articles from CNN with two views: news text and images in the news.
- Blogcatalog dataset [2]: A subset of blog posts from Blogcatalog website in the categories of {Autos, Software, Crafts, Football, Career&Jobs}. Two views are the text in posts and the tags associated with the posts, respectively.

The statistics of four real-world datasets is summarized in Table XII.

TABLE XIII: Clustering accuracy on four datasets. For the baselines that need parameter tuning, best/median performance is reported.

Method	**BBC Sport**				**Reuters**			
# features	100	200	300	400	100	200	300	400
All Features	0.5960				0.6545			
LS	0.4034	0.3885	0.3756	0.4112	0.3792	0.4587	0.5446	0.5900
UDFS	0.4565/0.4504	0.5232/0.5228	0.5549/0.5107	0.5525/0.5164	0.4320/0.4225	0.4921/0.4436	0.5926/0.4630	0.5918/0.5421
RSFS	0.6054/0.5388	0.6515/0.5709	0.6713/0.6041	0.6634/0.6085	0.5688/0.4558	0.5757/0.4529	0.6546/0.5271	0.6259/0.5332
MVFS	0.5996/0.5480	0.6572/0.5662	0.6148/0.5966	0.6118/0.6015	0.5302/0.4284	0.5561/0.4505	0.5592/0.5447	0.5950/0.5299
MVUFS	0.6253/0.4338	0.6181/0.5258	0.6242/0.6089	0.6542/0.6181	0.5998/0.3677	0.6476/0.4782	0.6397/0.5339	0.6182/0.5619
CDMA-FS	0.7341	0.7403	0.7472	0.7494	0.5465	0.6015	0.6322	0.6428
Method	**BlogCatalog**				**CNN**			
# features	100	200	300	400	100	200	300	400
All Features	0.5979				0.3005			
LS	0.3947	0.3975	0.4112	0.4550	0.2435	0.2419	0.2573	0.3238
UDFS	0.5219/0.4153	0.6173/0.6022	0.6561/0.6556	0.6489/0.6459	0.4095/0.4084	0.4019/0.3956	0.4171/0.3921	0.3962/0.3772
RSFS	0.6388/0.4995	0.6504/0.5733	0.6657/0.5917	0.6513/0.6014	0.3647/0.2692	0.4131/0.3140	0.4112/0.3608	0.4243/0.3596
MVFS	0.5409/0.5139	0.6027/0.5690	0.6107/0.5778	0.6457/0.6056	0.3578/0.2639	0.4204/0.3511	0.3902/0.3697	0.4213/0.3637
MVUFS	0.6157/0.4901	0.6693/0.6157	0.6565/0.5514	0.6496/0.5521	0.4524/0.3227	0.4899/0.3520	0.4879/0.3402	0.4649/0.3566
CDMA-FS	0.6029	0.6746	0.6704	0.6851	0.5347	0.4989	0.4771	0.4783

[1] https://sites.google.com/site/qianmingjie/home/datasets/cnn-and-fox-news

[2] http://dmml.asu.edu/users/xufei/datasets.html

TABLE XIV: Clustering NMI on four datasets. For the baselines that need parameter tuning, best/median performance is reported.

Method	**BBC Sport**				**Reuters**			
# features	100	200	300	400	100	200	300	400
All Features	0.4434				0.4846			
LS	0.0724	0.0775	0.0702	0.1099	0.1960	0.2689	0.3486	0.3989
UDFS	0.2279/0.1968	0.3453/0.2994	0.3453/0.2939	0.3386/0.2861	0.2203/0.2187	0.2829/0.2639	0.4023/0.2834	0.4046/0.3677
RSFS	0.3543/0.3141	0.4340/0.3900	0.5162/0.4151	0.5076/0.4166	0.4079/0.2429	0.4329/0.2963	0.4539/0.3648	0.4666/0.4134
MVFS	0.3383/0.3133	0.4288/0.3899	0.4276/0.4155	0.4371/0.4157	0.3594/0.2267	0.3986/0.2787	0.4256/0.3855	0.4427/0.4180
MVUFS	0.4374/0.2062	0.4255/0.3171	0.4273/0.4032	0.4443/0.4236	0.4260/0.1866	0.4887/0.3346	0.4816/0.3570	0.4681/0.4000
CDMA-FS	0.5774	0.6659	0.6693	0.6738	0.3823	0.4532	0.4801	0.4858
Method	**BlogCatalog**				**CNN**			
# features	100	200	300	400	100	200	300	400
All Features	0.4782				0.0957			
LS	0.2252	0.2458	0.2400	0.2819	0.0513	0.0557	0.0667	0.1280
UDFS	0.3223/0.1978	0.4123/0.3580	0.4501/0.4309	0.4753/0.4328	0.2122/0.1897	0.1852/0.1846	0.1920/0.1831	0.1868/0.1784
RSFS	0.4260/0.3090	0.4551/0.3564	0.4715/0.4064	0.4746/0.4408	0.1537/0.0690	0.1862/0.0984	0.1853/0.1430	0.2048/0.1383
MVFS	0.3432/0.3181	0.3971/0.3543	0.4274/0.4041	0.4764/0.4424	0.1517/0.0739	0.2051/0.1391	0.1558/0.1444	0.2034/0.1391
MVUFS	0.4237/0.2910	0.4747/0.4347	0.4643/0.3997	0.4504/0.3998	0.2242/0.1170	0.2824/0.1340	0.2917/0.1423	0.2670/0.1645
CDMA-FS	0.4176	0.4650	0.4866	0.5105	0.3244	0.3244	0.3049	0.2910

6.7.2 Baselines

We compare CDMA-FS with using all features and five other unsupervised feature selection methods as follows:

- All Features: It uses all original features without selection for evaluation.
- LS: Laplacian Score (34) selects the features that preserve the local manifold structure.
- UDFS: Unsupervised Discriminative Feature Selection (106) is a pseudo-label based approach with $L_{2,1}$ regularization to exploit the local structure.
- RSFS: Robust Spectral Feature Selection (76) selects features by robust spectral analysis framework with sparse regression.

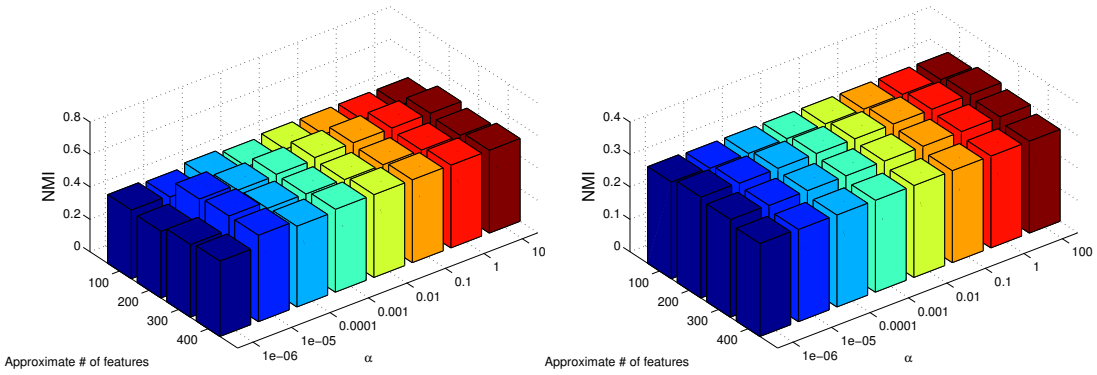
- MVFS: Multi-view Feature Selection (82) is unsupervised feature selection for multi-view data based on pseudo labels, which are generated as the consensus of spectral clustering on two views.
- MVUFS: Multi-view Unsupervised Feature Selection (69) generates pseudo-labels by Non-negative Matrix Factorization and local kernel learning.

6.7.3 Experiment setup

In this section, we evaluate the quality of selected features by their clustering performance. We use the the popular co-regularized spectral clustering (43) for clustering multi-view data [1]. We set their σ as the median of pairwise Euclidean distances between data points and $\lambda = 0.1$ as suggested in the paper. KMeans is then used on these latent factors. We repeat the KMeans experiment for 20 times (since it is initilization) and report the average performance. We vary the number of features d in the range of $\{100, 200, 300, 400\}$. For each feature size d, we choose appropriate λ in our method via binary search to let the number of selected features (with score $s_p = 1$) within $d \pm 10$.

Following the typical experimental setting for unsupervised feature selection (106) (48) (101), we use Accuracy and Normalized Mutual Information (NMI) to evaluate the result of clustering. We set $k = 5$ for the kNN neighbor size in the baseline methods and our approach following previous convention (48). For the number of pseudo-classes in UDFS, RSFS, MVFS and MVUFS, we use the ground-truth number of classes. Also, we perform grid search in the range of $\{0.1, 1, 10\}$ for the regularization parameters in these baseline methods. Besides their best performance, we also report the median performance for them. For CDMA-FS proposed in this paper, we use '0/1' weighting in the $\mathbf{W}$

[1] We use the code at `http://www.umiacs.umd.edu/~abhishek/code_coregspectral.zip`



(a) BlogCatalog (b) CNN

Figure 12: NMI w.r.t different values of α

and we fix $\sigma^2 = 1$ and $\alpha = 0.01$ for all the datasets after normalizing each data point to unit length. We set the maximum number of iterations for the cross-diffusion process as 20.

6.7.4 Results

The clustering accuracy and NMI on four datasets are shown in Table XIII and Table XIV. It can be observed that feature selection is a useful technique for improving the multi-view clustering performance. For example, compared with using all the features, CDMA-FS with 400 features improves the accuracy on BBCSport and BlogCatalog datasets by 26% and 15%, respectively. When comparing with other feature selection methods, we can observe that CDMA-FS performs favorably or comparable to the best performance of baseline methods, the parameters of which are tuned using all the class labels. Considering that in practice one cannot know the best parameters for these baseline methods (since

we assume no supervision), their median performance is a better reflection of these methods' practical power, which is far inferior to CDMA-FS.

6.7.5 Parameter Sensitivity

In this subsection, we study how the regularization α in the cross diffusion process affects the quality of selected features. The performance w.r.t different α on BlogCatalog and CNN is shown in Figure 12. We can observe that the performance is not very sensitive to α, and CDMA-FS can perform reasonably well when $\alpha > 10^{-5}$. In contrast, the baseline methods in Table XIII and Table XIV tend to be more sensitive w.r.t. the parameter values, as the their median performance differs significantly with best performance.

CHAPTER 7

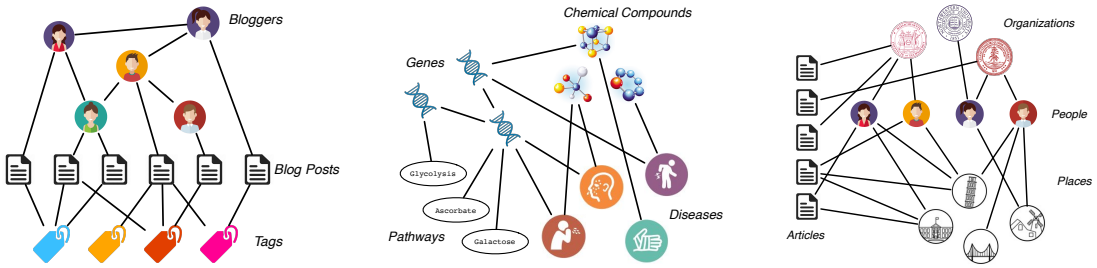
UNSUPERVISED FEATURE SELECTION WITH COMPLEX SIDE INFORMATION

7.1 Introduction

High dimensional data become increasingly popular as people are able to collect information from different aspects. For example, thousands of gene profiling features are obtained from microarray experiments, and data collected from sensors that are deployed at various locations across the country could compose feature vectors of more than one million entries.

Such high dimensionality poses challenges to many machine learning and data mining tasks because many features in the high dimensional space are noisy and irrelevant. Feature selection (34) (106), by removing noisy features and retaining a set of high-quality ones, can help alleviate the curse of dimensionality. In addition to improving the performance, selecting a small set of relevant features can also make the machine learning models more interpretable and provide additional insights about the problem.

Supervised feature selection (77) selects features by measuring their correlations with class labels which are usually expensive to obtain. Therefore, we mainly focus on unsupervised feature selection in this chapter. Unsupervised feature selection methods (34) (106) usually aim to exploit the statistics information of the data. However, using the potentially noisy features alone as guidance might be insufficient for selecting high-quality features.



(a) Blogs (b) Chemical compounds (c) News articles

Figure 13: Examples of data with complex side information

In the era of big data, one can often collect various forms of side information associated with the entities of interest. Such side information can usually provide abundant information about the data instances. Hence, to select high-quality features, it is highly desirable to incorporate the side information. However, side information usually comes in a complex and interrelated form and poses additional challenges on how to use it effectively (Figure 13).

- In blog websites (e.g., BlogCatalog), each blog can be represented as a high dimensional feature vector from the text data. Besides such text features, blog posts are also equipped with complex side information as shown in 13a where each blog post is associated with a user who writes the blog and a set of tags describing the post. In addition, there are social relationships between users. Considering the cost of supervised feature selection with labels from human experts, it would be a worthwhile effort to utilize the side information to guide feature selection.
- In bioinformatics, each chemical compound can be represented by its substructures in the vector space. Consider the task of predicting the side effect of drugs (i.e., chemical compounds)

based on these substructure features. The supervision information (i.e., side effect) can be very expensive to obtain through clinical trials (e.g., sometimes even at the cost of human lives), and thus supervised feature selection is less effective in this scenario. Fortunately, one can have a set of side information (e.g., gene-chemical compound interaction, gene-pathway interaction and gene-disease interaction, as in 13b) which provides rich information for selecting informative substructure features.

- In news articles, there are different concepts or entities, such as people, places and organizations. The relationships between these entities can be extracted from external knowledge bases, such as Freebase (9). For instance, people can be related with multiple places and organizations (13c). Such external knowledge can also be useful for guiding the selection of text features.

Side information can provide valuable information for feature selection, especially when class labels are unavailable. In this chapter, we study the problem of unsupervised feature selection with complex side information. To handle the increasingly complicated form of side information, we propose a new method, SideFS (Complex Side Information-guided Feature Selection), in this chapter. Since the side information is usually interrelated, we model them as a heterogeneous information network (79) (90) (47). We then derive similarity measures between instances based on the concept of meta-path. Information is derived from the meta-paths by learning network based representations. Such representations are used to guide feature selection. The contributions of this chapter can be summarized as follows.

- To our best knowledge, we are the first to formulate the problem of unsupervised feature selection with complex form of side information.

- We propose a novel method, SideFS, which performs joint feature selection and representations learning from the complex side information by modeling it as a heterogeneous information network.
- We conduct experiments on real-world datasets and show that SideFS can effectively utilize the side information and outperform baseline methods significantly.

7.2 Related Work

In this section, we briefly review unsupervised feature selection.

Unsupervised feature selection typically utilizes information only from the features by employing different criteria. One popular criterion principle is to preserve the local manifold structure or similarity (34) (109) (110). These unsupervised feature selection algorithms usually evaluate the importance of features individually and neglect correlation among features. Psuedo-label based methods (106) (48) (92) with $L_{2,1}$ norm (63) have gained much popularity in recent years. In these methods, sparse regression/subspace learning are performed to achieve the effect of feature selection. These methods mainly differ in the the way of generating pseudo-labels and the constraints on pseudo-labels. Non-negative Discriminative Feature Selection (NDFS) (48) enforces non-negative constraint on the latent factors obtained from spectral analysis. Robust Unsupervised Feature Selection (RUFS) (68) further adds $L_{2,1}$ norm to the regression objective to be robust to outlier instances. Robust Spectral Feature Selection (RSFS) (76) uses local kernel regression and robust Huber loss. FSASL (21) performs feature selection and local structure learning jointly. Recently, Stochastic Neighbor-preserving Feature Selection (SNFS) are proposed to jointly evaluate features in a non-linear manner based on the concept of stochastic neighbors (101).

However, these methods are not able to exploit side information in the complex form, which could contain abundant information.

7.3 Proposed Method

We denote n data samples as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ and the dimensionality of original feature space is D. So $\mathbf{x}_i \in \mathbb{R}^D$ and x_{ip} denotes the value of p-th ($p = 1, \ldots, D$) feature of $\mathbf{x}_i$. Our goal is to select d ($d \ll D$) high-quality features.

7.3.1 Knowledge Extraction from Complex Side Information

The first step of the framework is to extract knowledge from the complex side information. We model the complex relationship of entities in the side information as a heterogeneous information network. The key idea of this knowledge extraction step is to first derive meta-paths from the side information network and encode the side information via embedding learning.

We model the inter-connected objects in the complex side information as a heterogeneous *side information network*:

Definition 12 Side Information Network *The complex side information of data instances can be represented as a Side Information Network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ denotes the set of nodes, which includes t types of entities, $\mathcal{V}_1 = \{v_{11}, v_{12}, \ldots, v_{1n_1}\}, \ldots, \mathcal{V}_t = \{v_{t1}, v_{t2}, \ldots, v_{tn_t}\}$. $\mathcal{E}$ denotes the set of (multiple types of) links $\mathcal{E} \subset V \times V$.*

The target data instances are also one type of nodes in the side information network and we refer to them as **instance nodes**.

TABLE XV: Examples of meta-paths derived from two datasets

Datasets	Examples of meta-path
BlogCatalog	Blog $\xrightarrow{has}$ Tag $\xrightarrow{has^{-1}}$ Blog
	Blog $\xrightarrow{written\_by}$ User $\xrightarrow{written\_by^{-1}}$ Blog
	Blog $\xrightarrow{written\_by}$ User $\xrightarrow{friend}$ User $\xrightarrow{written\_by^{-1}}$ Blog
	Blog $\xrightarrow{written\_by}$ User $\xrightarrow{friend}$ User $\xrightarrow{friend}$ User $\xrightarrow{written\_by^{-1}}$ Blog
Chemical Compound	Compound $\xrightarrow{treat}$Disease$\xrightarrow{treat^{-1}}$Compound
	Compound$\xrightarrow{bind}$ Gene $\xrightarrow{bind^{-1}}$Compound
	Compound$\xrightarrow{bind}$ Gene $\xrightarrow{PPI}$ Gene $\xrightarrow{bind^{-1}}$Compound
	Compound$\xrightarrow{treat}$Disease$\xrightarrow{cause^{-1}}$Gene$\xrightarrow{bind^{-1}}$ Compound
	Compound $\xrightarrow{bind}$ Gene $\xrightarrow{has}$ Pathway $\xrightarrow{has^{-1}}$ Gene $\xrightarrow{bind^{-1}}$ Compound

Definition 13 Meta-path *A meta-path $\mathcal{P}$ of length l represents a sequence of relations $\mathcal{R}_i$ ($i = 1, \ldots, l$), i.e., $\mathcal{T}_1 \xrightarrow{\mathcal{R}_1} \mathcal{T}_2 \xrightarrow{\mathcal{R}_2} \cdots \xrightarrow{\mathcal{R}_l} \mathcal{T}_{l+1}$, where $\mathcal{T}_i$ ($i = 1, \ldots, l+1$) are the types of nodes. A unique sequence of nodes is referred to as a* **path instance** *of $\mathcal{P}$.*

For each pair of instances, various meta-paths can be extracted to provide information about their correlations (Table XV). Different types of meta-paths usually have different semantic meanings. For example, meta-path *Compound-Disease-Compound* means chemical compounds that can cure the same disease, while meta-path *Compound-Gene-Pathway-Gene-Compound* indicates chemical compounds binding with the genes that are involved in the same pathway.

Inspired by the path-counting measure in (80), we define the following side information-based similarity measure by counting the meta-path instances between the target data points.

Definition 14 SideSim *Given a side information network, we define the following similarity measure from the side information w.r.t meta-path $m \in M$ as follows:*

$$s_{ij}^{(m)} = \frac{2 \cdot |\mathcal{P}^{(m)}(i \rightsquigarrow j)|}{|\mathcal{P}^{(m)}(i \rightsquigarrow \cdot)| + |\mathcal{P}^{(m)}(j \rightsquigarrow \cdot)|} \tag{7.1}$$

where $|\mathcal{P}^{(m)}(i \rightsquigarrow j)|$ denotes the number of path instances with type m between data instances i and j, and $|\mathcal{P}^{(m)}(i \rightsquigarrow \cdot)|$ denotes the number of out-going path instances of type m from instance i.

These multiple types of meta-paths depict the correlations among target data instances from complementary perspectives, and it is desirable to ensemble them to obtain a more comprehensive view of correlations. We consider the following two ways of aggregation, which we refer to as **Micro Aggregation** and **Macro Aggregation**. We will compare the performance of these two aggregation methods in experiments.

Definition 15 Micro SideSim Aggregation *We define the micro-aggregation of SideSim as follows:*

$$s_{ij} = \frac{\sum_{m \in M} \cdot 2w^{(m)} |\mathcal{P}^{(m)}(i \rightsquigarrow j)|}{\sum_{m \in M} w^{(m)} |\mathcal{P}^{(m)}(i \rightsquigarrow \cdot)| + \sum_{m \in M} w^{(m)} |\mathcal{P}^{(m)}(j \rightsquigarrow \cdot)|} \tag{7.2}$$

Definition 16 Macro SideSim Aggregation *We define the macro-aggregation of SideSim as follows:*

$$s_{ij} = \sum_{m \in M} w^{(m)} s_{ij}^{(m)} \tag{7.3}$$

where $w^{(m)}$ is the weight assigned for meta-path with type m. In the unsupervised scenario, one could just use equal weights for all types of meta-paths, as the simplest form of ensemble. Alternatively,

one could rely on domain experts to provide prior knowledge to determine the importance of different meta-paths. We adopt the former approach in our experiments.

We further define the transition probability $\mathbf{P}$ based on the aggregated SideSim

$$P_{ij} = \frac{s_{ij}}{\sum_{j=1}^{n} s_{ij}} \tag{7.4}$$

Considering that the SideSim between instances are not quite reliable for non-nearest pairs, we truncate the fused full similarity graph to a kNN graph G^f based on $\hat{P} = (P_{ij} + P_{ij}^T)/2$ which tends to have better performance in our preliminary experiments. We use $k = 10$ in this chapter.

To further extract information from the fused graph G^f, we learn embeddings from this graph structure. Since the connected instances tend to have larger correlations, we learn the embeddings $\mathbf{u}_i \in \mathbb{R}^c$ $(i = 1, 2, \ldots, n)$ for each instance to make the embeddings of neighbors in G^f close and embeddings of non-neighbors far apart. Hence, we minimize the negative log-likelihood as follows:

$$\min_{\mathbf{U}} L^g = -\sum_{(i,j)\in E} \log\left(f_{ij}\right) - \sum_{(i,j)\in NE} \log\left(1 - f_{ij}\right) + \gamma||\mathbf{U}||_F^2 \tag{7.5}$$

where $\mathbf{U} = [\mathbf{u}_1^T, \ldots, \mathbf{u}_n^T]^T$ and γ controls the complexity of $\mathbf{U}$ ($||\cdot||_F$ denotes Frobenius norm). f_{ij} should be a monotonic function that transforms the similarity or distance between $\mathbf{u}_i$ and $\mathbf{u}_j$ into the range of $(0, 1)$. For example, f_{ij} could be $\frac{1}{1+\exp\left(-\mathbf{U}_i\mathbf{U}_j^T\right)}$ or $\frac{1}{1+||\mathbf{U}_i-\mathbf{U}_j||_F^2}$. We found these two functions have similar performance in our preliminary experiments and we use the former one in the rest of chapter. For the set of negative edges NE in Equation 7.5, we perform negative sampling as in (81) and retain $|E|$ number of negative edges.

7.3.2 Joint Representation Learning and Feature Selection

In the previous subsection, we discuss how to learn representations from the fused side information network. Side information could also be noisy, so the representations derived from the side information network might not be high-quality for every data instance. Under such scenario, it is desirable to incorporate information from the instance features for the representation learning. Meanwhile, we perform feature selection jointly in this process.

To utilize these representations for feature selection, we learn a linear projection of $\mathbf{U}$.

$$\min_{\mathbf{V}} ||\mathbf{U}\mathbf{V}^T - \mathbf{X}||_F^2 \tag{7.6}$$

A projection matrix $\mathbf{V} \in \mathbb{R}^{D \times c}$ is introduced to establish the connection between the representations $\mathbf{U}$ and the feature matrix $\mathbf{X}$ in Equation 7.6.

To perform joint feature selection when learning the representation, we employ a feature selection indicator vector $\mathbf{s} \in \{0,1\}^D$, where $s_p = 1$ indicates the p-th feature is selected and $s_p = 0$ otherwise.

$$\begin{aligned} \min_{\mathbf{V},\mathbf{s}} \quad & ||\mathbf{U}\mathbf{V}^T \operatorname{diag}(\mathbf{s}) - \mathbf{X}||_F^2 \\ \text{s.t. } & s_p \in \{0,1\}, \forall p = 1, \ldots, D \\ & \sum_{p=1}^{D} s_p = d \end{aligned} \tag{7.7}$$

where $\operatorname{diag}(\mathbf{s})$ is the diagonal matrix with $\mathbf{s}$ as the diagonal elements. The constraint $\sum_{p=1}^{D} s_p = d$ enforces that only d ($d < D$) features are retained. Intuitively, the representations leverage information from both the original features and the rich information from the side information. The features

that cannot be well represented by the latent representations through linear projection tend to be noisy features and will be removed. s_p of such features tend to be 0 under the constraint of $\sum_{p=1}^{D} s_p = d$. $\mathrm{diag}(\mathbf{s})\mathbf{V}$ is a matrix with d non-zero rows and hence it achieves the effect of feature selection.

We can combine $\mathbf{s}$ and $\mathbf{V}$ together, and employ $L_{2,0}$ norm to achieve the effect of feature selection. The $L_{2,0}$ norm $||\mathbf{W}||_{2,0}$ is the number of rows in $\mathbf{V}$ with non-zero values. If $||\mathbf{V}_{i\cdot}||_F = 0$, the i-th feature is not selected. The feasible region defined by $||\mathbf{V}||_{2,0} \leq d$ is not convex and we relax $||\mathbf{V}||_{2,0}$ into its convex hull $||\mathbf{V}||_{2,1}$:

$$\begin{aligned} \min_{\mathbf{V}} \ & ||\mathbf{U}\mathbf{V}^T - \mathbf{X}||_F^2 \\ \text{s.t. } & ||\mathbf{V}||_{2,1} \leq d \end{aligned} \tag{7.8}$$

where the $L_{2,1}$ norm $||\mathbf{V}||_{2,1} = \sum_{i=1}^{D} ||\mathbf{V}_{i\cdot}||_F$ could also achieve row sparsity. We further write the constraint in the form of Lagrangian as follows:

$$\min_{\mathbf{V}} L^a = ||\mathbf{U}\mathbf{V}^T - \mathbf{X}||_F^2 + \lambda ||\mathbf{V}||_{2,1} \tag{7.9}$$

where λ is the regularization parameter on $L_{2,1}$ norm.

We combine the side information-based loss and feature-based loss together, and the final objective function becomes the following:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} L &= L^g + L^a \\ &= -\sum_{(i,j)\in E} \log\left(f_{ij}\right) - \sum_{(i,j)\in NE} \log\left(1 - f_{ij}\right) + \\ &\quad \gamma||\mathbf{U}^g||_F^2 + \alpha||\mathbf{U}\mathbf{V}^T - \mathbf{X}||_F^2 + \lambda||\mathbf{V}||_{2,1}\end{aligned} \tag{7.10}$$

where α is the parameter that controls the relative importance of consensus learning.

7.4 Optimization

In this section, we discuss how to solve the optimization problem for SideFS.

The objective function in Equation 7.10 is not jointly convex on both $\mathbf{U}$ and $\mathbf{V}$. Now we decompose the objective function into two subproblems and develop an alternating optimization approach to solve the problem in Equation 7.10.

Step 1. Fixing $\mathbf{V}$, update $\mathbf{U}$. With fixed $\mathbf{V}$, we optimize the following objective:

$$\begin{aligned}\min_{\mathbf{U}} L_u &= L_u^g + L_u^a + L_u^{reg} \\ L_u^g &= -\sum_{(i,j)\in E} \log\left(f_{ij}\right) - \sum_{(i,j)\in NE} \log\left(1 - f_{ij}\right) \\ L_u^a &= \alpha||\mathbf{U}\mathbf{V} - \mathbf{X}||_F^2 \\ L_u^{reg} &= \gamma||\mathbf{U}||_F^2\end{aligned}$$

It is easy to verify this objective function is convex w.r.t $\mathbf{U}$ when $\mathbf{V}$ is fixed. The following gradients can be derived w.r.t. $\mathbf{U}$.

$$\frac{\partial L_u^g}{\partial \mathbf{u}_i} = \sum_{(i,j)\in E} (f_{ij} - 1) \cdot \mathbf{u}_j + \sum_{(i,j)\in NE} f_{ij} \cdot \mathbf{u}_j \tag{7.11}$$

$$\frac{\partial L_u^a}{\partial \mathbf{U}} = 2(\mathbf{UV} - \mathbf{X})\mathbf{V} \tag{7.12}$$

To solve this subproblem, one can use gradient-based methods, such as L-BFGS.

Step 2. Fixing $\mathbf{U}$, update $\mathbf{V}$. With fixed $\mathbf{U}$, we find the optimal $\mathbf{V}$ by solving the following problem:

$$\min_{\mathbf{V}} L_v = ||\mathbf{UV} - \mathbf{X}||_F^2 + \frac{\lambda}{\alpha}||\mathbf{V}||_{2,1} \tag{7.13}$$

We denote $L_v^{reg} = \frac{\lambda}{\alpha}||\mathbf{V}||_{2,1}$. We can derive the following gradient w.r.t. $\mathbf{V}$:

$$\frac{\partial L_v}{\partial \mathbf{V}} = 2(\mathbf{UV} - \mathbf{X})' \cdot \mathbf{U} + \frac{\partial L_v^{reg}}{\partial \mathbf{V}} \tag{7.14}$$

$$\frac{\partial L_v^{reg}}{\partial V_{ij}} = \frac{\lambda}{\alpha} \frac{V_{ij}}{\sqrt{\sum_{j=1}^{k} V_{ij} + \epsilon}} \tag{7.15}$$

where ϵ is a very small positive number that makes the regularization term differentiable at 0 (105).

We perform step 1 and 2 in alternating manner, as shown in Algorithm 7.

Theorem 7.4.1 *For the optimization problem in Equation 7.10, Algorithm 7 is guaranteed to converge.*

Algorithm 7 Alternating Optimization for SideFS

Initialize: $\mathbf{U} = rand(0, 1)$, $\mathbf{V} = \mathbf{0}$, $t = 1$.
while not converged **do**
  Fixing $\mathbf{V}$, find the optimal $\mathbf{U}$ by L-BFGS
  Fixing $\mathbf{U}$, find the optimal $\mathbf{V}$ by L-BFGS
  $t = t + 1$
end while
Output: Rank all the features ($i = 1, \ldots, D$) by $||\mathbf{V}_{i\cdot}||$ and return the top d features.

TABLE XVI: Statistics of two datasets

Statistics	BlogCatalog	Chemical Compound
# of instances	3083	105
# of features	3170	290
# of labels	5	550

Proof: The objective function in Equation 7.10 monotonically decreases in each iteration and it is lowered bounded. So the alternating framework in Algorithm 7 would converge.

7.5 Experiments

In this section, we compare the proposed method with several baselines with applications on clustering and multi-label prediction.

7.5.1 Datasets

- BlogCatalog Dataset[1]: A subset of blog post dataset in the following categories: {Personal Development, Investing, Fitness, Soccer, Cars}. A blog post can have side information such as users, tags and relationships between users.
- Chemical Compound Dataset (42): A bioinformatics network in which each chemical compound has subgraph features mined from the compound structure. Besides, we also have complex side information such as genes, diseases, pathways and PPIs (protein-protein interactions).

The statistics of these two datasets is shown in Table XVI and the interrelated complex side information is illustrated in 13a and 13b.

7.5.2 Baselines

We compare our method to the following unsupervised feature selection methods.

- All Features: It uses all original features without selection for evaluation.
- LS: Laplacian Score (34) selects the features which can best preserve the local manifold structure.
- UDFS: Unsupervised Discriminative Feature Selection (106) is a pseudo-label based approach with $L_{2,1}$ regularization.
- RSFS: Robust Spectral Feature Selection (76) selects features by $L_{2,1}$-norm regularized regression with robust Huber loss.

---

[1] `http://dmml.asu.edu/users/xufei/datasets.html`

TABLE XVII: Clustering performance on BlogCatalog

Accuracy (All Features: 0.6224)						
# of Features	LS	UDFS	RSFS	SNFS	Micro-SideFS	Macro-SideFS
100	0.2809	0.4490	0.3973	0.6103	**0.6796**	0.6740
200	0.3730	0.5489	0.5292	0.6670	0.7284	**0.7333**
300	0.3958	0.6147	0.5752	0.6840	**0.7247**	0.7157
400	0.4311	0.6380	0.6059	0.6020	**0.7430**	0.7351
NMI (All Features: 0.4667)						
# of Features	LS	UDFS	RSFS	SNFS	Micro-SideFS	Macro-SideFS
100	0.0193	0.2113	0.1453	0.3870	0.4595	**0.4671**
200	0.1195	0.3391	0.3268	0.4673	**0.5348**	0.5252
300	0.1503	0.4346	0.3837	0.5000	**0.5387**	0.5377
400	0.2109	0.4451	0.4298	0.4357	**0.5628**	0.5570

- SNFS: The recently proposed Stochastic Neighbor-preserving Feature Selection (101) evaluates features jointly in a non-linear manner.

For all methods (except SNFS), we do grid search for the regularization parameter in the range of $\{0.1, 1, 10\}$ and report the best performance. For SNFS, we follow the author's suggestion and choose the λ that makes $N_{0.9}$ close to the desired number of features. We use $c = 5$ as the latent dimension size in our method and the baselines. For the proposed SideFS, we use all the non-redundant meta-paths with length less than 5, since previous work (79) suggests meta-paths with large length tend to be not as useful.

7.5.3 Clustering Blog Posts

For the BlogCatalog dataset, we evaluate the feature quality by the clustering performance. We use Accuracy and Normalized Mutual Information (NMI) as evaluation metrics following the convention in

existing work (106) (68) (76). The cluster labels are mapped to ground truth labels using Kuhn-Munkres Algorithm (59). Normalized Mutual Information (NMI) is defined based on the mutual information between cluster labels and class labels. Higher values of accuracy and NMI indicate better quality of clustering. For all methods, we perform K-means[1] on the selected features. Since K-means is affected by the initial seeds, we repeat the experiment for 20 times and report the average performance.

Results The clustering performance is shown in Table XVII. When comparing the two variants of SideFS, Macro-SideFS and Micro-SideFS performs similarly with different numbers of features. Compared with other feature selection methods, both Micro-SideFS and Macro-SideFS outperform the baseline methods significantly with different feature sizes. For example, the best performance achieved by Micro-SideFS and Macro-SideFS, which can effectively utilize side information, improves clustering accuracy over the most competitive baseline SNFS by 8.6% and 7.5%, respectively. This suggests both micro and macro aggregation methods can be effective to incorporate side information.

7.5.4 Predicting Side Effect of Chemical Compounds

In this subsection, we evaluate the feature quality by their performance in predicting side effects for chemical compounds. Selecting informative substructures can help human experts develop better insights on the mechanisms of compound structures and their potential risks on incurring side effects.

Similar to previous work (12), we use 1-NN as the classifier for the prediction task. Since a chemical compound might cause more than one side effects, we use the micro-F1, macro-F1 and Hamming Loss as the performance measures. The performance of different methods is shown in Table XVIII. The

---

[1]We use the code at `http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html`

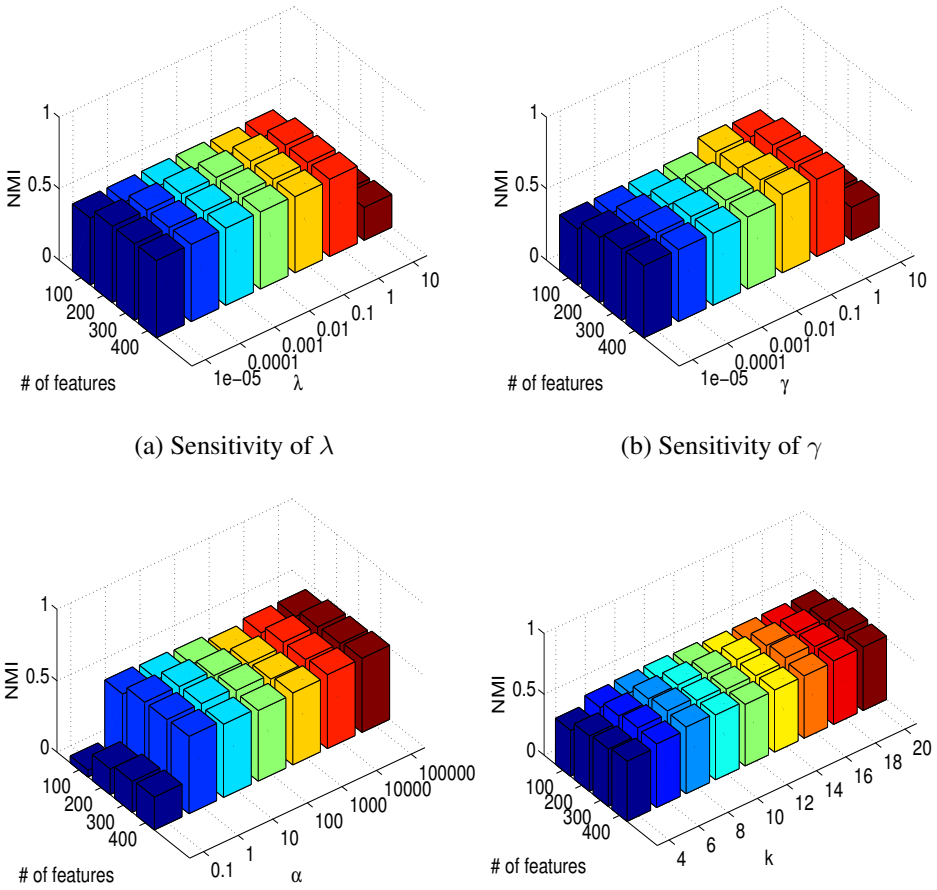
TABLE XVIII: 1-NN performance on side effect prediction. ↑ indicates that larger value is better while ↓ indicates that smaller value is better. The best result on each metric is in bold font.

Micro-F1 ↑ (All Features: 0.0913)					
LS	UDFS	RSFS	SNFS	Micro-SideFS	Macro-SideFS
0.0799	0.0866	0.0936	0.0825	**0.1041**	0.0978
Macro-F1 ↑ (All Features: 0.1061)					
LS	UDFS	RSFS	SNFS	Micro-SideFS	Macro-SideFS
0.0946	0.1023	0.1094	0.1029	**0.1177**	0.1127
Hamming Loss ↓ (All Features: 0.0456)					
LS	UDFS	RSFS	SNFS	Micro-SideFS	Macro-SideFS
0.0477	0.0459	0.0456	0.0477	0.0453	**0.0431**

features selected by SideFS usually outperform baseline methods by $5\% \sim 10\%$. This illustrates the usefulness of incorporating the side information into the feature selection process.

7.5.5 Sensitivity Analysis

In our algorithm, there are several regularization parameters λ, α, γ and latent dimension size c. In this subsection, we investigate how the proposed method performs under different values of parameters (vary one parameter when fixing $c = 5$ and others equal to 1) with feature sizes $\{100, 200, 300, 400\}$. The NMI results on the BlogCatalog dataset with micro-aggregation are shown in Figure 14. We can observe that SideFS is not very sensitive to these parameters and performs well for a wide range of parameter values.



(a) Sensitivity of λ

(b) Sensitivity of γ

(c) Sensitivity of α

(d) Sensitivity of k

Figure 14: Parameter sensitivity w.r.t. different parameters

CHAPTER 8

CONCLUSIONS AND CONTRIBUTION

(Parts of chapter were previously published (101), (99), (96), (100) and (98).)

In this thesis, we have explored unsupervised feature selection for heterogeneous forms of data. Towards this direction, we thoroughly studied different forms of data: traditional data, network data, multi-view data and data with complex side information. We presented methods related to feature selection on these types of data with novel perspectives and insights. The effectiveness of the proposed approaches are evaluated by experiments on various real-world datasets. The contributions we have made can be summarized as follows:

- We propose a new method, SNFS, for unsupervised feature selection by preserving stochastic neighbors. For each data point, other data points can be its potential neighbors with certain probability. We select the features that can approximate the original distribution by minimizing the KL-divergence. This criterion can select discriminative features that makes similar data points close and push dissimilar data points far apart. The objective function has less parameters than the state-of-the-art pseudo-label methods and it has a simple gradient update formula. We develop an efficient optimization algorithm for SNFS based on projected L-BFGS. Empirical results show that the proposed method outperforms state-of-the-art approaches on several real-world datasets.
- Network structures present valuable information as well as new challenges to feature selection. We develop an efficient unsupervised feature selection algorithm for network data based on partial

order preserving (POP) principle, a new perspective on using links to guide feature selection. Our method is conceptually simple and computationally efficient, whereas state-of-the-art approaches typically involve heavy matrix computation and are intractable for large real world networks. Experiments indicate that our approach significantly outperforms state-of-the-art methods in terms of both efficiency and effectiveness.

- We develop an unsupervised feature selection algorithm from a generative point of view which can incorporate information from the node content and links directly. We assume that the node attributes and link structures are generated from a set of oracle features and we aim to recover this set of high-quality features based on the generation process. Experiments indicate that our approach significantly outperforms state-of-the-art methods in terms of feature quality.
- In many real-world networks, the links and node attributes are often partially observable. We study how to recommend link-only nodes to attribute-only nodes (or vice versa), by learning a representation consensus between links and attributes. Two instantiations that employ different ranking-based loss are presented for the representation learning. Considering high-dimensional node attributes are potentially noisy, we perform joint feature selection in the representation learning process. The link-based representation and the attribute-based representation could lend strength to each other and make the representation more resilient to link and attribute noise. Experimental results shows that the proposed P-NRCL and MM-NRCL are able to learn high-quality representations, which can effectively perform cross-view link prediction.
- High-dimensional multi-view data pose challenges for many machine learning tasks. While feature selection methods can be useful for alleviating the curse of dimensionality, existing ap-

proaches either cannot exploit information from multiple views simultaneously or rely on cluster labels for this task. We aim to preserve more accurate information from multi-view data by learning a cross-diffused matrix and directly utilize the information. Experimental results show that CDMA-FS is able to select high-quality features on real-world datasets and outperforms the baseline methods significantly.

- By observing many datasets are equipped with complex side information, we propose a novel method, SideFS, for unsupervised feature selection with heterogeneous side information. Such side information can provide useful information when the class labels are expensive to obtain We leverage the side information by learning representations from the meta-paths and such representations can be used to guide feature selection. Experimental results show that incorporating side information can effectively enhance the quality of selected features in real-world applications.

CITED LITERATURE

1. L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

2. L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.

3. N. Barbieri, F. Bonchi, and G. Manco. Who to follow and why: link prediction with explanations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1266–1275, 2014.

4. J. Barzilai and J. M. Borwein. Two-Point Step Size Gradient Methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

5. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2002.

6. D. P. Bertsekas. Projected newton methods for optimization problems with simple constraints. In *SIAM Jounal on Control and Optimization*, 1982.

7. E. G. Birgin, J. M. Martnez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000.

8. A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

9. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA, 2008.

10. F. Bonchi, C. Castillo, A. Gionis, and A. Jaimes. Social network analysis and mining for business applications. *ACM Trans. Intell. Syst. Technol.*, 2(3):22:1–22:37, 2011.

11. D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized non-negative matrix factorization for data representation. *PAMI*, 33(8):1548–1560, 2011.

12. D. Cai, C. Zhang, and X. He. Unsupervised feature selection for multi-cluster data. In *KDD*, pages 333–342, 2010.

13. P. Calamai and J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39:93–116, 1987.

14. B. Cao, L. He, X. Wei, M. Xing, P. S. Yu, H. Klumpp, and A. D. Leow. t-bne: Tensor-based brain network embedding. In *SDM*, 2017.

15. B. Cao, C.-T. Lu, X. Wei, P. S. Yu, and A. D. Leow. Semi-supervised tensor factorization for brain network analysis. In *ECML/PKDD*, 2016.

16. Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, pages 129–136, 2007.

17. G. C. Cawley, N. L. C. Talbot, and M. Girolami. Sparse multinomial logistic regression via bayesian l1 regularisation. In *NIPS*, pages 209–216, 2006.

18. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

19. C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13:795–828, 2012.

20. N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S. Kandola. On kernel-target alignment. In *NIPS*, pages 367–373, 2001.

21. L. Du and Y.-D. Shen. Unsupervised feature selection with adaptive structure learning. In *KDD*, pages 209–218, 2015.

22. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, 2 edition, 2001.

23. J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, 2004.

24. M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

25. Y. Feng, J. Xiao, Y. Zhuang, and X. L. 0002. Adaptive unsupervised multi-view feature selection for visual concept recognition. In *ACCV (1)*, volume 7724, pages 343–357, 2012.

26. E. M. Gafni and D. P. Bertsekas. Two-metric projection methods for constrained optimization. In *SIAM Journal on Control and Optimization*, 1984.

27. L. Grippo, F. Lampariello, and S. Lucidi. A Nonmonotone Line Search Technique for Newton's Method. *SIAM Journal on Numerical Analysis*, 23(4):707–716, 1986.

28. L. Grippo and M. Sciandrone. On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Operations Research Letters*, 26:127–136, 2000.

29. A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.

30. Q. Gu and J. Han. Towards feature selection in network. In *CIKM*, pages 1175–1184, 2011.

31. Q. Gu and J. Zhou. Local learning regularized nonnegative matrix factorization. In *IJCAI*, pages 1046–1051, 2009.

32. J. Gui, D. Tao, Z. Sun, Y. Luo, X. You, and Y. Y. Tang. Group sparse multiview patch alignment framework with view consistency for image classification. *IEEE Transactions on Image Processing*, 23(7):3126–3137, 2014.

33. M. A. Hasan and M. J. Zaki. A survey of link prediction in social networks. In *Social Network Data Analytics*, pages 243–275. 2011.

34. X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *NIPS*, 2005.

35. G. Hinton and S. Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15:833–840, 2003.

36. D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *KDD*, pages 593–598, 2004.

37. T. Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

38. T. Joachims. Training linear svms in linear time. In *KDD*, pages 217–226, 2006.

39. T. Joachims, T. Finley, and C.-N. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

40. I. Kahanda and J. Neville. Using transactional information to predict link strength in online social networks. In *ICWSM*, 2009.

41. L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, VOL. 18, NO. 1:39– 43, 1953.

42. X. Kong, B. Cao, and P. S. Yu. Multi-label classification by mining label and instance correlations from heterogeneous information networks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 614–622, 2013.

43. A. Kumar, P. Rai, and H. D. III. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

44. J. Kunegis and A. Lommatzsch. Learning spectral graph transformations for link prediction. In *ICML*, volume 382, page 71. ACM, 2009.

45. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

46. M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

47. H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao. Spotting fake reviews via collective positive-unlabeled learning. In *ICDM*, 2014.

48. Z. Li, Y. Yang, J. Liu, X. Zhou, and H. Lu. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*, 2012.

49. D. Liben-Nowell and J. M. Kleinberg. The link prediction problem for social networks. In *CIKM*, pages 556–559, 2003.

50. R. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 243–252, 2010.

51. W. Lin, X. Kong, P. S. Yu, Q. Wu, Y. Jia, and C. Li. Community detection in incomplete information networks. In *Proceedings of the 21st International Conference on World Wide Web*, pages 341–350. ACM, 2012.

52. D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

53. H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In *In Proceedings of 7th IEEE Int'l Conference on Tools with Artificial Intelligence*, 1995.

54. L. L and T. Zhou. Link prediction in complex networks: A survey. *Physica A*, 390(6):1150–1170, 2011.

55. B. McFee and G. R. G. Lanckriet. Partial order embedding with multiple kernels. In *ICML*, volume 382, page 91, 2009.

56. A. K. Menon and C. Elkan. Link prediction via matrix factorization. In *Proceedings of the ECML/PKDD 2011*, 2011.

57. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

58. K. T. Miller, T. L. Griffiths, and M. I. Jordan. Nonparametric latent feature models for link prediction. In *NIPS*, pages 1276–1284, 2009.

59. J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, 1957.

60. M. E. Newman. Modularity and community structure in networks. *Proc Natl Acad Sci U S A*, 103(23):8577–8582, 2006.

61. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, February 2004.

62. A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

63. F. Nie, H. Huang, X. Cai, and C. H. Q. Ding. Efficient and robust feature selection via joint 2, 1-norms minimization. In *NIPS*, pages 1813–1821, 2010.

64. H. Peng, F. Long, and C. H. Q. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238, 2005.

65. B. Perozzi, R. Al-Rfou', and S. Skiena. Deepwalk: online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710. ACM, 2014.

66. O. Perron. Zur Theorie der Matrices. *Mathematische Annalen*, 64(2):248–263, 1907.

67. A. Popescul and L. H. Ungar. Statistical relational learning for link prediction. In *IJCAI03 Workshop on Learning Statistical Models from Relational Data*, 2003.

68. M. Qian and C. Zhai. Robust unsupervised feature selection. In *IJCAI*, 2013.

69. M. Qian and C. Zhai. Unsupervised feature selection for multi-view clustering on text-image web news data. In *CIKM*, pages 1963–1966. ACM, 2014.

70. S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI*, 2009.

71. M. Schmidt, E. V. D. Berg, M. P. Friedl, and K. Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. In *In AI & Statistics*, 2009.

72. M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *NIPS*, 2004.

73. P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

74. S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, volume 227, pages 807–814, 2007.

75. W. Shao, L. He, C.-T. Lu, X. Wei, and P. S. Yu. Online unsupervised multi-view feature selection. In *ICDM*, 2016.

76. L. Shi, L. Du, and Y.-D. Shen. Robust spectral learning for unsupervised feature selection. In *ICDM*, 2014.

77. E. J. Smola, S. V. N. Vishwanathan, and Q. Lenicta. Bundle methods for machine learning. In *NIPS*, 2008.

78. L. Song, A. J. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *ICML*, volume 227, pages 823–830. ACM, 2007.

79. Y. Sun, J. Han, C. C. Aggarwal, and N. V. Chawla. When will it happen?: relationship prediction in heterogeneous information networks. In *WSDM*, pages 663–672. ACM, 2012.

80. Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, 2011.

81. J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, 2015.

82. J. Tang, X. Hu, H. Gao, and H. Liu. Unsupervised feature selection for multi-view data in social media. In *SDM*, pages 270–278. SIAM, 2013.

83. J. Tang and H. Liu. Feature selection with linked data in social media. In *SDM*, pages 118–128, 2012.

84. J. Tang and H. Liu. Unsupervised feature selection for linked social media data. In *KDD*, pages 904–912, 2012.

85. J. Tang and H. Liu. Coselect: Feature selection with instance selection for social media data. In *SDM*, pages 695–703. SIAM, 2013.

86. L. Tang and H. Liu. Relational learning via latent social dimensions. In *KDD*, 2009.

87. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

88. B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu. Unsupervised metric fusion by cross diffusion. In *CVPR*, pages 2997–3004, 2012.

89. C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *ICDM*, pages 322–331, 2007.

90. C. Wang, Y. Song, H. Li, M. Zhang, and J. Han. Knowsim: A document similarity measure on structured heterogeneous information networks. In *ICDM*, pages 1015–1020, 2015.

91. F. Wang, T. Li, X. Wang, S. Zhu, and C. H. Q. Ding. Community discovery using nonnegative matrix factorization. *Data Min. Knowl. Discov.*, 22:493–521, 2011.

92. S. Wang, J. Tang, and H. Liu. Embedded unsupervised feature selection. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 470–476, 2015.

93. W. Wang and Z.-H. Zhou. A new analysis of co-training. In *ICML*, pages 1135–1142, 2010.

94. X. Wei, B. Cao, W. Shao, C.-T. Lu, and P. S. Yu. Community detection with partially observable links and node attributes. In *IEEE International Conference on Big Data*, 2016.

95. X. Wei, B. Cao, and P. S. Yu. Nonlinear joint unsupervised feature selection. In *SDM*, 2016.

96. X. Wei, B. Cao, and P. S. Yu. Unsupervised feature selection on networks: A generative view. In *AAAI*, 2016.

97. X. Wei, B. Cao, and P. S. Yu. Unsupervised feature selection with complex side information. In *manuscript*, 2016.

98. X. Wei, B. Cao, and P. S. Yu. Multi-view unsupervised feature selection by cross-diffused matrix alignment. In *IJCNN*, 2017.

99. X. Wei, S. Xie, and P. S. Yu. Efficient partial order preserving unsupervised feature selection on networks. In *SDM*, pages 82–90. SIAM, 2015.

100. X. Wei, L. Xu, B. Cao, and P. S. Yu. Cross view link prediction by learning noise-resilient representation consensus. In *WWW*, 2017.

101. X. Wei and P. S. Yu. Unsupervised feature selection by preserving stochastic neighbors. In *AISTATS*, 2016.

102. L. Xu, X. Wei, J. Cao, and P. S. Yu. Embedding identity and interest for social networks. In *WWW*, 2017.

103. L. Xu, X. Wei, J. Cao, and P. S. Yu. Embedding of embedding (eoe): Joint embedding for coupled heterogeneous networks. In *WSDM*, pages 741–749, 2017.

104. L. Xu, X. Wei, J. Cao, and P. S. Yu. On learning mixed community-specific similarity metrics for cold-start link prediction. In *WWW*, 2017.

105. L. Yan, W.-J. Li, G.-R. Xue, and D. Han. Coupled group lasso for web-scale ctr prediction in display advertising. In *ICML*, volume 32, pages 802–810, 2014.

106. Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou. l2, 1-norm regularized discriminative feature selection for unsupervised learning. In *IJCAI*, pages 1589–1594, 2011.

107. K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu. Stochastic relational models for discriminative link prediction. In *NIPS*, pages 1553–1560, 2006.

108. J. Zhang, P. S. Yu, and Z.-H. Zhou. Meta-path based multi-network collective link prediction. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1286–1295, 2014.

109. Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, volume 227, pages 1151–1157, 2007.

110. Z. Zhao, L. Wang, and H. Liu. Efficient spectral feature selection with minimum redundancy. In *AAAI*, 2010.

111. Z. Zhao, L. Wang, H. Liu, and J. Ye. On similarity preserving feature selection. *IEEE Trans. Knowl. Data Eng.*, 25(3):619–632, 2013.

112. M. Zheng, J. Bu, C. Chen, C. Wang, L. Z. 0005, G. Qiu, and D. Cai. Graph regularized sparse coding for image representation. *IEEE Transactions on Image Processing*, 20(5):1327–1336, 2011.

VITA

Name: Xiaokai Wei

EDUCATION:

M.Sc. in Mathematics, University of Illinois at Chicago, 2016

B.Eng. in Computer Science, Beijing University of Posts and Telecommunications, 2010.

PUBLICATIONS:

- **Xiaokai Wei**, Bokai Cao and Philip S. Yu. Multi-view Unsupervied Feature Selection by Cross-diffused Matrix Alignment. To appear in **IJCNN 2017**
- Bokai Cao, Lifang He, **Xiaokai Wei**, Mengqi Xing, Philip S. Yu, Heide Klumpp and Alex D. Leow. *t-BNE: Tensor-based Brain Network Embedding*. To appear in **SDM 2017**
- Linchuan Xu, **Xiaokai Wei**, Jiannong Cao and Philip S. Yu. On Learning Mixed Community-specific Similarity Metrics for Cold-start Link Prediction. To appear in **WWW 2017** (poster)
- Linchuan Xu, **Xiaokai Wei**, Jiannong Cao and Philip S. Yu. Embedding Identity and Interest for Social Networks. To appear in **WWW 2017** (poster)
- **Xiaokai Wei**, Linchuan Xu, Bokai Cao and Philip S. Yu. Cross-view Link Prediction with Noise-resilient Representation Learning. To appear in **WWW 2017**
- Linchuan Xu, **Xiaokai Wei**, Jiannong Cao and Philip S. Yu. *Embedding of Embedings (EOE): Embedding for Coupled Heterogeneous Networks.* **WSDM 2017**
- **Xiaokai Wei**, Bokai Cao, Weixiang Shao, Chun-Ta Lu and Philip S. Yu. *Community Detection with Partially Observable Links and Node Attributes*. **IEEE Big Data 2016**

- Weixiang Shao, Lifang He, Chun-Ta Lu, **Xiaokai Wei** and Philip Yu. *Online Unsupervised Multi-view Feature Selection*. **ICDM 2016**
- Bokai Cao, Chun-Ta Lu, **Xiaokai Wei**, Philip S. Yu and Alex D. Leow. *Semi-supervised Tensor Factorization for Brain Network Analysis*. **ECML/PKDD 2016**
- **Xiaokai Wei** and Philip S. Yu. *Unsupervised Feature Selection by Preserving Stochastic Neighbors*. **AISTATS 2016**
- **Xiaokai Wei**, Bokai Cao and Philip S. Yu. *Nonlinear Joint Unsupervised Feature Selection*. **SDM 2016**
- **Xiaokai Wei**, Bokai Cao and Philip S. Yu. *Unsupervised Feature Selection on Networks: A Generative View*. **AAAI 2016**
- **Xiaokai Wei**, Sihong Xie and Philip S. Yu. *Efficient Partial-Order Preserving Unsupervised Feature Selection on Networks*. **SDM 2015**
- Huayi Li, Zhiyuan Chen, Bing Liu, **Xiaokai Wei** and Jidong Shao. *Spotting Fake Reviews via Collective Positive-Unlabeled Learning*. **ICDM 2014**

You hereby grant to the Journal of Machine Learning Research the following rights with respect to the article described below:

- exclusive right of first publication in all media
- right of subsequent publication in all media
- the right to sublicense or transfer this license to any publisher of Journal of Machine Learning Research.

This license will be valid throughout the world throughout the entire term of copyright in the article, and is granted free of royalty. You will ensure that any dissemination of the article authorized by you gives appropriate first publication credit to Journal of Machine Learning Research.

You explicitly reserve all other proprietary rights with respect to the article, including copyright and patent rights.

If there is more than one author of the article, the word "you" includes all authors jointly and severally. By signing this form, the signatory acknowledges that he or she is the corresponding author; that the signatory is signing on behalf of all authors of the article; and that the signatory has the authority to act as the agent for the other authors of the article.

You warrant that you are the sole author of the article, and generally that you have a complete and unencumbered right to make the grants you make to us. You also warrant that the article does not libel anyone, invade anyone's copyright or otherwise violate any statutory or common law right of anyone, and that you have made all reasonable efforts to ensure the accuracy of any factual information contained in the article. You agree to indemnify us and any sublicensees against any claim or action alleging facts which, if true, constitute a breach of any of the foregoing warranties.

This is the entire agreement between us, and it may be modified only in writing. It will be governed by the laws of the Commonwealth of Massachusetts. It will bind and benefit our respective assigns and successors in interest, including your heirs. It will terminate if we do not publish, in any medium, your article within two years of the date of your signature(s).

Title of Article Unsupervised Feature Selection by Preserving Stochastic Neighbors

Author name(s) Xiaokai Wei
Philip S. Yu

Name of Signatory Xiaokai Wei

Signature Xiaokai Wei

Date April 18th, 2016

In order for SIAM to include your paper in the 2015 SIAM International Conference on Data Mining proceedings, the following Copyright Transfer Agreement must be agreed to during the paper upload process.

COPYRIGHT TRANSFER AGREEMENT

Title of Paper: Efficient Partial Order Preserving Unsupervised Feature Selection on Networks

Author(s): Xiaokai Wei, Sihong Xie, Philip S. Yu

Paper ID Number: 619

Copyright to this paper is hereby irrevocably assigned to SIAM for publication in the **Proceedings of the SIAM International Conference on Data Mining (SDM15)**, April 30 – May 2, 2015 at the Renaissance Vancouver Harbourside Hotel, Vancouver, British Columbia, Canada. SIAM has sole use for distribution in all forms and media, such as microfilm and anthologies, except that the author(s) or, in the case of a "work made for hire," the employer will retain:

> The right to use all or part of the content of the paper in future works of the author(s), including author's teaching, technical collaborations, conference presentations, lectures, or other scholarly works and professional activities as well as to the extent the fair use provisions of the U.S. Copyright Act permit. If the copyright is granted to SIAM, then the proper notice of the SIAM's copyright should be provided.
>
> The right to post the final draft of the paper on noncommercial pre-print serves like arXiv.org.
>
> The right to post the final version of the paper on the author's personal web site and on the web server of the author's institution, provided the proper notice of the SIAM's copyright is included and that no separate or additional fees are collected for access to or distribution of the paper.
>
> The right to refuse permission to third parties to republish all or part of the paper or translation thereof.

It is affirmed that neither this paper nor portions of it have been published elsewhere and that a copyright transfer agreement has not been signed permitting the publication of a similar paper in a journal or elsewhere. For multi-author works, the signing author agrees to notify all co-authors of his/her action.

Copyright Release
[X] Yes, I agree to the terms of the SIAM copyright.

Work Made for Hire
[__] Check here if signature is on behalf of employer in the event article is "work made for hire."

Previously Published

Check here if portions have been published elsewhere and enclose appropriate credits and permissions to republish.
[__] Yes (if yes, expand site so authors can enclose credits and permissions)
[__] No

Alternate Copyright
[__] Check here to submit an alternative copyright. Send copyright to erle@siam.org and asen@siam.org using subject line "SIAM – SDM15 – Alternate Copyright – LAST NAME."

Association for the Advancement of Artificial Intelligence
2275 East Bayshore Road, Suite 160
Palo Alto, California 94303 USA

AAAI COPYRIGHT FORM

Title of Article/Paper: Unsupervised Feature Selection on Networks: A Generative View

Publication in Which Article/Paper Is to Appear: 30th AAAI Conference on Artificial Intelligence (AAAI-16)

Author's Name(s): Xiaokai Wei, Bokai Cao, Philip S. Yu

Please type or print your name(s) as you wish it (them) to appear in print

PART A – COPYRIGHT TRANSFER FORM

The undersigned, desiring to publish the above article/paper in a publication of the Association for the Advancement of Artificial Intelligence, (AAAI), hereby transfer their copyrights in the above article/paper to the Association for the Advancement of Artificial Intelligence (AAAI), in order to deal with future requests for reprints, translations, anthologies, reproductions, excerpts, and other publications.

This grant will include, without limitation, the entire copyright in the article/paper in all countries of the world, including all renewals, extensions, and reversions thereof, whether such rights current exist or hereafter come into effect, and also the exclusive right to create electronic versions of the article/paper, to the extent that such right is not subsumed under copyright.

The undersigned warrants that they are the sole author and owner of the copyright in the above article/paper, except for those portions shown to be in quotations; that the article/paper is original throughout; and that the undersigned right to make the grants set forth above is complete and unencumbered.

If anyone brings any claim or action alleging facts that, if true, constitute a breach of any of the foregoing warranties, the undersigned will hold harmless and indemnify AAAI, their grantees, their licensees, and their distributors against any liability, whether under judgment, decree, or compromise, and any legal fees and expenses arising out of that claim or actions, and the undersigned will cooperate fully in any defense AAAI may make to such claim or action. Moreover, the undersigned agrees to cooperate in any claim or other action seeking to protect or enforce any right the undersigned has granted to AAAI in the article/paper. If any such claim or action fails because of facts that constitute a breach of any of the foregoing warranties, the undersigned agrees to reimburse whomever brings such claim or action for expenses and attorneys' fees incurred therein.

Returned Rights

In return for these rights, AAAI hereby grants to the above author(s), and the employer(s) for whom the work was performed, royalty-free permission to:

1. Retain all proprietary rights other than copyright (such as patent rights).
2. Personal reuse of all or portions of the above article/paper in other works of their own authorship. This does not include granting third-party requests for reprinting, republishing, or other types of reuse. AAAI must handle all such third-party requests.
3. Reproduce, or have reproduced, the above article/paper for the author's personal use, or for company use provided that AAAI copyright and the source are indicated, and that the copies are not used in a way that implies AAAI endorsement of a product or service of an employer, and that the copies per se are not offered for sale. The foregoing right shall not permit the posting of the article/paper in electronic or digital form on any computer network, except by the author or the author's employer, and then only on the author's or the employer's own web page or ftp site. Such web page or ftp site, in addition to the aforementioned requirements of this Paragraph, shall not post other AAAI copyrighted materials not of the author's or the employer's creation (including tables of contents with links to other papers) without AAAI's written permission.
4. Make limited distribution of all or portions of the above article/paper prior to publication.
5. In the case of work performed under a U.S. Government contract or grant, AAAI recognized that the U.S. Government has royalty-free permission to reproduce all or portions of the above Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract or grant so requires.

In the event the above article/paper is not accepted and published by AAAI, or is withdrawn by the author(s) before acceptance by AAAI, this agreement becomes null and void.

(1) Xiaokai Wei — Nov 30, 2015

Author/Authorized Agent for Joint Author's Signature — Date

N/A — Title (if not author)

Employer for whom work was performed

(For jointly authored Works, all joint authors should sign unless one of the authors has been duly authorized to act as agent for the others.)

Association for the Advancement of Artificial Intelligence
2275 East Bayshore Road, Suite 160
Palo Alto, California 94303 USA

PART B – U.S. GOVERNMENT EMPLOYEE CERTIFICATION

This will certify that all authors of the above article/paper are employees of the U.S. Government and performed this work as part of their employment, and that the article/paper is therefore not subject to U.S. copyright protection. The undersigned warrants that they are the sole author/translator of the above article/paper, and that the article/paper is original throughout, except for those portions shown to be in quotations.

(2)

U.S. Government Employee Authorized Signature _____ Date _____

Name of Government Organization _____ Title (if not author) _____

*(Please read and sign and return Part B **only** if you are a government employee and created your article/paper as part of your employment. If your work was performed under Government contract, but you are not a Government employee, sign only at signature line (1) in Part A and see item 5 under returned rights. Authors who are U.S. government employees should also sign signature line (1) in Part A above to enable AAAI to claim and protect its copyright in international jurisdictions.)*

PART C–CROWN COPYRIGHT CERTIFICATION

This will certify that all authors of the above article/paper are employees of the British or British Commonwealth Government and prepared the Work in connection with their official duties , and that the article/paper is therefore subject to Crown Copyright and is not assigned to AAAI as set forth in the first sentence of the Copyright Transfer Section in Part A. The undersigned warrants that they are the sole author/translator of the above article/paper, and that the article/paper is original throughout, except for those portions shown to be in quotations, and acknowledges that AAAI has the right to publish, distribute, and reprint the Work in all forms and all media.

(3)

British or British Commonwealth Government Employee Authorized Signature _____ Date _____

Name of Government Organization _____ Title (if not author) _____

*(Please read and sign and return Part C **only** if you are a British or British Commonwealth Government employee and the Work is subject to Crown Copyright. Authors who are British or British Commonwealth government employees should also sign signature line (1) in Part A above to indicate their acceptance of all terms other than the copyright transfer.)*

IW3C2 Copyright Release Form

Title of work:	**Cross View Link Prediction by Learning Noise-resilient Representation Consensus**
Author(s):	**Xiaokai Wei, Linchuan Xu, Bokai Cao and Philip S. Yu**
Description of material:	**WWW Paper**
Title of IW3C2 Publication:	**WWW '17: 26th International World Wide Web Conference**

I hereby assign (to the extend transferable, see point B below) to the International World Wide Web Conference Committee (IW3C2) the copyright of this Work for the full period of copyright and all renewals, extensions, revisions and revivals together with all accrued rights of action throughout the world in any form, including as part of IW3C2 and the public Conference Web site, on CD-ROM and in translation, or on videocassette, broadcast, cablecast, laserdisc, multimedia or any other media format now or hereafter known. (Not all forms of media will be utilized.) I accept that IW3C2 will allow the Association for Computing Machinery (ACM) to distribute, make available on the Internet or sell this Material as part of the above-named publication in the ACM Digital Library. Finally, I also accept that IW3C2 will publish this Work under the Creative Commons Attribution 4.0 International (CC BY 4.0) license, which reserves my rights to disseminate the work on my personal and corporate Web site with the appropriate attribution. Notwithstanding the above, I retain all proprietary rights other than copyright as assigned above, such as patent and trademark rights. For further details, see the Appendix below.

In the event that any elements used in the Material contain the work of third-party individuals, I understand that it is my responsibility to secure any necessary permissions and/or licenses and will provide it in writing to IW3C2. If the copyright holder requires a citation to a copyrighted work, I have obtained the correct wording and have included it in the designated space in the text.

I hereby release and discharge IW3C2 and other publication sponsors and organizers from any and all liability arising out of my inclusion in the publication, or in connection with the performance of any of the activities described in this document as permitted herein. This includes, but is not limited to, my right of privacy or publicity, copyright, patent rights, trade secret rights, moral rights, or trademark rights.

All permissions and releases granted by me herein shall be effective in perpetuity and throughout the universe, and extend and apply to the IW3C2 and its assigns, contractors, sub-licensed distributors, successors, and agents.

The following statement of copyright ownership will be displayed with the Material, unless otherwise specified: **" © [year] International World Wide Web Conference Committee (IW3C2), published under Creative Commons CC By 4.0 License." IW3C2 reserves the right to provide a hyperlink to the author's site if the Material is used in electronic media.**

(●) A. I hereby represent and warrant that I am the sole owner (or authorized agent of the copyright owner(s)) OR

( ) B. I do not own some rights to this work: (check applicable)

- [ ] I have used third-party material and have permission to do so.
- [ ] I am employed by the national government of my country, prepared this work as part of my job and my work is not subject to copyright.
- [ ] I am employed by a body corporate, prepared this work as part of my job and my work is not subject to copyright.

DATE: **02/19/2017**

Appendix

WWW 2017, April 3-7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052575

IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

TITLE OF PAPER/ARTICLE/REPORT, INCLUDING ALL CONTENT IN ANY FORM, FORMAT, OR MEDIA (hereinafter, "the Work"):

854 Multi-view Unsupervised Feature Selection by Cross-diffused Matrix Alignment

COMPLETE LIST OF AUTHORS:

Xiaokai Wei, Bokai Cao and Philip S. Yu

IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book):

2017 International Joint Conference On Neural Networks (IJCNN)

COPYRIGHT TRANSFER

1. The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to: (a) the above Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia components or other enhancements accompanying the Work.

CONSENT AND RELEASE

2. In the event the undersigned makes a presentation based upon the Work at a conference hosted or sponsored in whole or in part by the IEEE, the undersigned, in consideration for his/her participation in the conference, hereby grants the IEEE the unlimited, worldwide, irrevocable permission to use, distribute, publish, license, exhibit, record, digitize, broadcast, reproduce and archive, in any format or medium, whether now known or hereafter developed: (a) his/her presentation and comments at the conference; (b) any written materials or multimedia files used in connection with his/her presentation; and (c) any recorded interviews of him/her (collectively, the "Presentation"). The permission granted includes the transcription and reproduction of the Presentation for inclusion in products sold or distributed by IEEE and live or recorded broadcast of the Presentation during or after the conference.

3. In connection with the permission granted in Section 2, the undersigned hereby grants IEEE the unlimited, worldwide, irrevocable right to use his/her name, picture, likeness, voice and biographical information as part of the advertisement, distribution and sale of products incorporating the Work or Presentation, and releases IEEE from any claim based on right of privacy or publicity.

4. The undersigned hereby warrants that the Work and Presentation (collectively, the "Materials") are original and that he/she is the author of the Materials. To the extent the Materials incorporate text passages, figures, data or other material from the works of others, the undersigned has obtained any necessary permissions. Where necessary, the undersigned has obtained all third party permissions and consents to grant the license above and has provided copies of such permissions and consents to IEEE.

☐ Please check this box if you do not wish to have video/audio recordings made of your conference presentation.

See reverse side for Retained Rights/Terms and Conditions, and Author Responsibilities.

GENERAL TERMS

- The undersigned represents that he/she has the power and authority to make and execute this assignment.
- The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
- In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall become null and void. Even in this case, IEEE will retain an archival copy of the manuscript.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.

(1) Xiaokai Wei _____ 02/27/2017

Author/Authorized Agent for Joint Authors **Date**

U.S. GOVERNMENT EMPLOYEE CERTIFICATION (WHERE APPLICABLE)

This will certify that all authors of the Work are U.S. government employees and prepared the Work on a subject within the scope of their official duties. As such, the Work is not subject to U.S. copyright protection.

(2) _____ _____

Authorized Signature **Date**

(Authors who are U.S. government employees should also sign signature line (1) above to enable the IEEE to claim and protect its copyright in international jurisdictions.)

CROWN COPYRIGHT CERTIFICATION (WHERE APPLICABLE)

This will certify that all authors of the Work are employees of the British or British Commonwealth Government and prepared the Work in connection with their official duties. As such, the Work is subject to Crown Copyright and is not assigned to the IEEE as set forth in the first sentence of the Copyright Transfer Section above. The undersigned acknowledges, however, that the IEEE has the right to publish, distribute and reprint the Work in all forms and media.

(3) _____ _____

Authorized Signature **Date**

(Authors who are British or British Commonwealth Government employees should also sign line (1) above to indicate their acceptance of all terms other than the copyright transfer.)

rev. 042016

IEEE COPYRIGHT FORM *(continued)*

RETAINED RIGHTS/TERMS AND CONDITIONS

General

1. Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.
2. Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.
3. In the case of a Work performed under a U.S. Government contract or grant, the IEEE recognizes that the U.S. Government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract/grant so requires.
4. Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use. The IEEE Intellectual Property Rights office must handle all such third-party requests.
5. Authors whose work was performed under a grant from a government funding agency are free to fulfill any deposit mandates from that funding agency.

Author Online Use

6. Personal Servers. Authors and/or their employers shall have the right to post the accepted version of IEEE-copyrighted articles on their own personal servers or the servers of their institutions or employers without permission from IEEE, provided that the posted version includes a prominently displayed IEEE copyright notice and, when published, a full citation to the original IEEE publication, including a link to the article abstract in IEEE Xplore. Authors shall not post the final, published versions of their papers.
7. Classroom or Internal Training Use. An author is expressly permitted to post any portion of the accepted version of his/her own IEEE-copyrighted articles on the author's personal web site or the servers of the author's institution or company in connection with the author's teaching, training, or work responsibilities, provided that the appropriate copyright, credit, and reuse notices appear prominently with the posted material. Examples of permitted uses are lecture materials, course packs, e-reserves, conference presentations, or in-house training courses.
8. Electronic Preprints. Before submitting an article to an IEEE publication, authors frequently post their manuscripts to their own web site, their employer's site, or to another server that invites constructive comment from colleagues. Upon submission of an article to IEEE, an author is required to transfer copyright in the article to IEEE, and the author must update any previously posted version of the article with a prominently displayed IEEE copyright notice. Upon publication of an article by the IEEE, the author must replace any previously posted electronic versions of the article with either (1) the full citation to the IEEE work with a Digital Object Identifier (DOI) or link to the article abstract in IEEE Xplore, or (2) the accepted version only (not the IEEE-published version), including the IEEE copyright notice and full citation, with a link to the final, published article in IEEE Xplore.

INFORMATION FOR AUTHORS

Author Responsibilities

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements as stated in section 8.2.1 of the IEEE PSPB Operations Manual, including provisions covering originality, authorship, author responsibilities and author misconduct. More information on IEEE's publishing policies may be found at http://www.ieee.org/publications_standards/publications/rights/authorrightsresponsibilities.html. Authors are advised especially of IEEE PSPB Operations Manual section 8.2.1.B12: "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it." Authors are also advised of IEEE PSPB Operations Manual section 8.1.1B: "Statements and opinions given in work published by the IEEE are the expression of the authors."

Author/Employer Rights

If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IEEE assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

IEEE Copyright Ownership

It is the formal policy of the IEEE to own the copyrights to all copyrightable material in its technical publications and to the individual contributions contained therein, in order to protect the interests of the IEEE, its authors and their employers, and, at the same time, to facilitate the appropriate re-use of this material by others. The IEEE distributes its technical publications throughout the world and does so by various means such as hard copy, microfiche, microfilm, and electronic media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various compendiums, collective works, databases and similar publications.

THIS FORM MUST ACCOMPANY THE SUBMISSION OF THE AUTHOR'S MANUSCRIPT.
Questions about the submission of the form or manuscript must be sent to the publication's editor.
Please direct all questions about IEEE copyright policy to:
IEEE Intellectual Property Rights Office, copyrights@ieee.org, +1-732-562-3966 (telephone)