**The Pennsylvania State University**
**The Graduate School**

# SEMI-SUPERVISED CLUSTERING FOR HIGH-DIMENSIONAL AND SPARSE FEATURES

A Dissertation in
Information Sciences and Technology
by
Su Yan



Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2010




The dissertation of Su Yan was reviewed and approved* by the following:

Dongwon Lee
Associate Professor of
Information Sciences and Technology
Dissertation Advisor, Co-Chair of Committee

C. Lee Giles
the David Reese Professor of
Information Sciences and Technology
Co-Chair of Committee

Jesse Barlow
Professor of
Computer Science and Engineering

Tracy Mullen
Assistant Professor of
Information Sciences and Technology

Carleen Maitland
Associate Professor of
Information Sciences and Technology

*Signatures are on file in the Graduate School.

# Abstract

Clustering is one of the most common data mining tasks, used frequently for data organization and analysis in various application domains. Traditional machine learning approaches to clustering are fully automated and unsupervised where class labels are unknown a priori. In real application domains, however, some "weak" form of side information about the domain or data sets can be often available or derivable. In particular, information in the form of instance-level pairwise constraints is general and is relatively easy to derive. The problem with traditional clustering techniques is that they cannot benefit from side information even when available. I study the problem of semi-supervised clustering, which aims to partition a set of unlabeled data items into coherent groups given a collection of constraints. Because semi-supervised clustering promises higher quality with little extra human effort, it is of great interest both in theory and in practice.

Semi-supervised clustering shares a difficulty with a large number of other learning methods in data mining literature. That is, they lose their algorithmic effectiveness for high dimensional data. I focus on data with high-dimensional sparse features and present a series of novel semi-supervised clustering approaches that are

both effective and efficient in learning from high-dimensional data. The proposed approaches are based on the dimensionality reduction idea. High-dimensional input data are embedded into an optimal low-dimensional subspace determined with the help of side information. The clustering structure of data is more evident in the subspace than in the original input space, and thus enable higher quality clustering solutions. The proposed clustering approaches explore both a small set of constraints and the large amount of unlabeled data, thus perform robustly even with limited side information. Besides, I also study how to automatically generate constraints based on domain knowledge. Since automatically generated constraints are inevitably noisy, I propose a semi-supervised approach that is able to use noisy side information to improve clustering accuracy. Moreover, the non-linear separability problem is studied in the semi-supervised clustering setting. I propose a solution that is computationally as easy as a linear-transformation based method, but is still able to separate non-linear data effectively.

# Table of Contents

# List of Figures

# List of Tables

# Acknowledgments

I am indebted to many people for all the help and support I received during my Ph.D. study and research.

First and foremost, I would like to thank my advisor, Dongwon Lee. Dr. Lee has been a constant source of motivation, ideas and guidance. He was patient and gave me a lot of freedom when I was stumbling around looking for a research problem in the early years. He was always accessible and willing to help me with research and more. He encouraged me and told me to trust myself whenever anything went wrong. Dr. Lee's passion about science, combination of intellectual depth and breadth, and great personality make my research life in Penn. State a smooth and rewarding experience.

I would like to thank my thesis committee members. Dr. Giles brought me into the world of data mining and machine learning. He has been a great mentor and collaborator. Dr. Barlow gave me great help on mathematics. I am always impressed by his mathematical vigor and sharp thinking. Dr. Mullen has been a constant source of help and is always willing to spend time discussing with me. I also thank Dr. Magy Seif El-Nasr. She has been a great advisor and I appreciate her creativity. It has been fun and enriching to work with her during my first 2 years at Penn. State.

My sincere thanks also goes to Steve Chen at IBM Sillicon Valley Lab, where I stayed for two summers, Dr. Ankur Jain and Dr. Daniel Nikovski at Mitsubishi

Electric Research Laboratories (MERL), where I stayed for half a year. I appreciate them for offering me internship opportunities in their groups and introducing me to many interesting projects.

I thank my friends, colleagues and student colleagues for inspirations, insightful comments, and numerous discussions that made up an important part of my Ph.D. life. They are my fellow labmates in the PIKE group: Dr. Byungon Won, Dr. Ergin Elmacioglu, Hung-sik Kim and Dr. Bo Luo, my student colleagues: Bingjun Sun, Dr. Issac Councill, Dr. Yang Sun, Bi Chen, Baojun Qiu, Qijun Pan, Yoon-Chan Jhi, Scott Robertson, Dr. Joseph A. Zupko, Dr. Kun Bai, Huajing Li, and Jing Chong, my friends meet during internship: Xiaoqian Jiang, Sofien Bouaziz, Mingyu Liu, Da Wang, Meng Wang, Dr. Oncel Tuzel, Dr. Kevin W. Wilson Dr. Yige Wang, Iren Liu, Dr. Mengchu Cai, and Dr. Xiaohong Fu.

I wish to thank Robin Kuzu from the University Office of Global Programs. Her patients and help make any complicated situation easier to handle.

Last and most importantly, I wish to thank my family, my mom Guifen Zhao, my dad Qingyi Yan, my husband Dr. Hai Wang, my sister Dr. Han Yan, and her husband Dr. Oliver Kruger. I thank them for helping me get through the difficult times, for all the emotional support and caring.

Chapter 1

# Introduction

Clustering, as a fundamental machine learning technique, is essential for exploratory data analysis in statistics, pattern recognition, information retrieval, data mining, and other fields. The study of clustering has a long history and a large number of approaches have been developed. However, significant challenges still remain. For example, like other learning techniques, a clustering approach often loses algorithmic effectiveness when handling high-dimensional data with sparse features. Moreover, given a data set to analyze, how to pick a proper clustering criteria is a difficult decision. In this thesis, I propose novel semi-supervised clustering approaches that are able to analyze high-dimensional data efficiently and effectively.

## 1.1 What is Clustering?

Clustering is a typical unsupervised machine learning task. The field of machine learning has traditionally been divided into three subfields

- Supervised Learning: The learning system is presented with data items $\mathbf{x}_i$ and explicit feedback of output values $\mathbf{y}_i$ in the form of input pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}|_{i=1}^{n}$. If $\mathcal{X}$ denotes the space of input values, and $\mathcal{Y}$ denotes the space of output values, the goal of supervised learning is to to learn a prediction function $h : \mathcal{X} \mapsto \mathcal{Y}$ so that the system makes good predictions $\{h(\mathbf{x}')\}|_{j=1}^{m}$ to new observed items $\{\mathbf{x}'_j\}|_{j=1}^{m}$. If output values $\mathbf{y}_i$ take discrete values, it is a *classification* problem, otherwise, a *regression* problem.

- Unsupervised Learning: The learning system is presented with data items $\{\mathbf{x}_i\}|_{i=1}^{n}$ only without feedback. Identifying the intrinsic structure and organization of a data set is the main goal of learning. For this reason, clustering is the supporting technique for data visualization, outlier detection, image segmentation, topic extraction from text corpus, and many more applications.

- Reinforcement Learning: The learning system has a set of environment states $\mathbb{S}$ and can take a set of actions $\mathbb{A}$. The learner receives a late feedback in the form of scalar reward for taking an action in state $s_t \in \mathbb{S}$ at time step $t$. The goal of learning is for the system to develop a scheme $h : \mathbb{S} \mapsto \mathbb{A}$ that takes

actions to yield the most reward.

## 1.2 What is Semi-Supervised Learning?

Semi-supervised learning is a new machine learning technique that has been attracting more and more research interest in recent years [Vap98], [NG00], [ZGL03], [ZG09]. In general, semi-supervised learning is the learning task where the learner observes data items $\{\mathbf{x}_i\}|_{i=1}^{n}$ and *partial feedback*. Semi-supervised learning problems fall into two major categories

- Semi-supervised classification/regression: As in supervised learning, the learning system observes a set of pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}|_{i=1}^{n}$ of data items with feedback in the form of predefined output values. However, the number of such item-and-feedback pairs is small. Besides, the system also observes a large number of data items without feedback $\{\mathbf{x}_j\}|_{j=1}^{m}$, $m \gg n$. Usually, a supervised learning system is unable to learn an accurate relationship between the space of input values $\mathcal{X}$ and the space of output values $\mathcal{Y}$ with a small amount of training data. Semi-supervised classification/regression, however, is able to learn from only a small amount of training data by also exploring the structure of unlabeled data $\{\mathbf{x}_j\}_{j=1}^{m}$.

- Semi-supervised clustering: Besides a set of unlabeled data items $\{\mathbf{x}_i\}|_{i=1}^{m}$, the learning system also observes *side information* $S$ taking various forms.

For example, the side information can say that items $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar, items $\mathbf{x}_p$ and $\mathbf{x}_q$ are different, or a cluster can contain no more than $m$ data items, etc. The side information serves as "weak supervision" to the learning system. So the learning task is different from unsupervised clustering, where the learning system cannot benefit from side information even available.

In this thesis, I focus on semi-supervised clustering problems. The side information I study takes the form of pairwise constraints. In particular, there are two types of pairwise constraints

- **Must-links**: two data items $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar and thus should be clustered together.
- **Cannot-links**: two data items $\mathbf{x}_i$ and $\mathbf{x}_j$ are different and thus should be placed into different clusters.

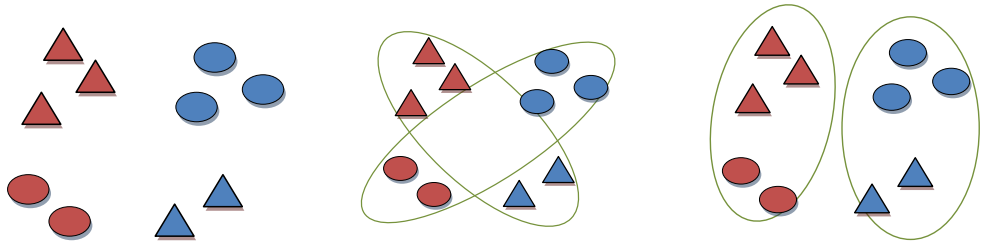
Side information in the form of pairwise constraints is general. For example, labeled training data can be expressed by pairwise constraints but not for inverse. Moreover, pairwise constraints naturally originate from many real application domains.

## 1.3 Why is Semi-Supervised Clustering Useful?

The side information that can be explored by semi-supervised clustering techniques originates naturally in many real application domains. Traditional unsupervised

clustering techniques are unable to benefit from the side information even available. For example, consider the problem of clustering human faces that appear in a video. A person's position does not change drastically during a video shot. For special videos, such as news or interviews, a person's position does not change much even during several video shots [CMM03] . It is reasonable to consider two faces that appear in adjacent frames in roughly the same position being the same person. For another example, clustering techniques have been used to solve name disambiguation problems [ETY+07] [HZG05]. Two name entities with similar spelling can be candidates for pairwise constraints. If two name entities share a large number of similar topics, or two author name entities share most of their coauthors, the two entities can be considered as referring to one person, even the spellings of the two entities have tiny difference which may due to spelling mistakes and other reasons. On the other hand, if they share no common topics, two entities with the same spelling can be considered as referring to two different people. Moreover, in the task of Web document clustering, documents which share a large number of similar hyperlinks, or a group of documents with strong co-citation (i.e., co-reference) patterns can be viewed as similar. Such domain-originated side information can provide supervision to the clustering process.

Even when domain knowledge is not available to generate side information, constraints can be derived manually with the help of human users. For example, consider the problem of clustering images of objects or human faces. It may be



(a) data items (b) clustering by shape 1 (c) clustering by color

Figure 1.1: One set of items has multiple reasonable clustering solutions.

difficult or costly for users to hand-label a large amount of images into pre-set class labels. However, when users are presented with a simple binary question of "are objects/faces in image $x_1$ and image $x_2$ the same object/person or not?", answering Yes/No to the question is a lot easier.

Moreover, the clustering solutions to a given set of items are often not unique. Consider the example shown in Figure 1.1. The data items can be reasonably clustered into two groups by color or by shape. The side information provided by a user reflects his/her expectation of the clustering results. Thus, semi-supervised clustering can explore the clustering criteria implicitly suggested by the side information, and can cluster items in a way that better satisfies a user's need.

## 1.4 Difficulties in Handling High-Dimensional Data

High-dimensional data are prevalent in applications such as database, text mining, image processing, sensor data analysis, and bioinformatics. For example, for

database with high-dimensional data, many indexing techniques construct a summary of the data set using a linear transformation scheme to reduce dimensions, and use the low dimensional synopsis for fast, approximate search.

Learning from high-dimensional data involves high computation cost. Besides, a learning system has the *curse of dimensionality problem* [Bel61] [Don00]. In particular, as the number of features keeps increasing, the learning performance can decrease after a certain point. This high-dimensionality difficulty has frequently been observed in practice [Ver03]. In supervised learning of high-dimensional data, the number of training data items is much less than the number of parameters to be learned. The learned model has high variance and does not generalize well. For unsupervised learning, as the feature dimensionality increases, data points become increasingly "sparse" [SEK03]. Thus, data items in the high dimensional space are equally far apart from each other no matter whether they are from the same cluster or not. Since all the clustering approaches critically rely on pairwise distances between data items, many clustering techniques lose their algorithmic effectiveness when dealing with high-dimensional data.

It is often necessary to reduce the dimensionality when dealing with high-dimensional data. *Feature selection* and *feature reduction* are two ways to reduce dimensionality.

- Feature selection reduces dimensionality by selecting a subset of existing features. Thus, the physical interpretation of each feature is preserved in the

reduced space. One may apply judicious feature selection to greatly reduce the number of features prior to learning from the data. The effects of front-end selection in supervised text classification were considered in [MN98]. The results demonstrated that in removing many features, information about the underlying data groups may be lost. Besides, a criterion function for feature selection is typically defined as a function of the classification error. Thus, feature selection is mainly used in supervised learning. For clustering tasks, since labels are not available, selecting an appropriate subset of features is difficult.

- Feature reduction reduces dimensionality by combining features with linear or non-linear transformations. Feature reduction is applicable to both supervised learning and unsupervised learning, depending on the availability of training data. A feature reduction approach can greatly reduce the feature space dimensionality while still preserve discriminative information. However, unlike in feature selection where the selected features retain their original physical interpretation, the new features generated by a feature reduction approach usually do not have a clear physical meaning.

In general, the choice between feature reduction and feature selection depends on the application domain. Since this thesis is about clustering, I focus on feature reduction algorithms because they are also applicable to unsupervised learning.

The feature reduction techniques can be linear or non-linear. Linear approaches

are fast and suitable for practical application. However, when data lie in a complicated manifold, non-linear feature reduction algorithms are able to represent data better in the reduced space. In this thesis, I focus on linear approaches due to their practicability. I will also show a new feature reduction approach that is linear but still has the advantages of non-linear methods.

At last, it is also important to note that high-dimensional spaces also have their advantages. Interesting readers refer to [Don00] for details.

## 1.5 Problem Definition

Let $\mathcal{X}$ be the input space containing $n$ data points in $f$ dimensions, $\{\mathbf{x}_i\}_{i=1}^n \in \mathcal{X}$. We are given two types of pairwise constraints organized in two sets. Let $\Omega_M = \{(\mathbf{x}_i, \mathbf{x'}_i)\}_{i=1}^m$ be the set of $m$ pairs of must-link constraints, and $\Omega_C = \{(\mathbf{x}_i, \mathbf{x'}_i)\}_{i=1}^c$ be the set of $c$ pairs of cannot-link constraints. Let $r$ be a desired subspace dimensionality. The goal is to embed the $f$-dimensional data in an $r$-dimensional subspace, s.t. $r \ll f$, by learning a linear data transformation $\mathbf{Z} \in \mathbb{R}^{f \times r}$, such that $\mathbf{y} = \mathbf{Z}^T\mathbf{x}$ where $\mathbf{y}$ is the low-dimensional embedding of $\mathbf{x}$. The main research tasks focus on learning the data transformation $\mathbf{Z}$ by exploring pairwise constraints $\Omega_M$ and $\Omega_C$, as well as exploring the structure of input data $\{\mathbf{x}_i\}_{i=1}^n$. After the transformation $\mathbf{Z}$ is learned, the Euclidean distance between two points $\mathbf{y}_1$ and $\mathbf{y}_2$ in the reduced space can be expressed as

$$d(\mathbf{y}_1, \mathbf{y}_2) = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{Z}\mathbf{Z}^T (\mathbf{x}_1 - \mathbf{x}_2)} \tag{1.1}$$

which only depends on the original data points and the learned transformation matrix.

## 1.6 Structure of the Thesis

The thesis is organized as follows. In Chapter 2, I briefly provide the background of clustering and feature reduction techniques, and summarize representative clustering and feature reduction approaches that are closely related to my thesis work. In Chapter 3, I review existing work on semi-supervised learning in general, semi-supervised clustering in particular, and put a focus on feature-reduction-based approaches. In Chapter 4, I focus on clustering efficiency, and introduce a simple semi-supervised dimension reduction approach that significantly improves clustering efficiencies and accuracies. In Chapter 5, I focus on automatically identifying side information from domain knowledge, and introduce a semi-supervised clustering approach that is able to explore noisy side information. In Chapter 6, I explore the problem of separating non-linear-separable data with linear transformations. At last, in Chapter 7, I conclude my thesis work and discuss interesting future work directions.

## 1.7 Notations

Through out the thesis, I use the following notation conventions.

- An *important concept* is emphasized with Italic font in the first mention.
- A matrix $\mathbf{Y}$ is denoted by a boldfaced capital letter.
- A vector $\mathbf{y}$ is denoted by a boldfaced lowercase letter.
- A scalar $y$ is denoted by an ordinary lowercase letter.
- A set $\mathbb{Y}$ is denoted by this special font.
- A function or operation is denoted by an ordinary capital or lowercase letter followed by parentheses, e.g., $Y(\cdot)$ or $y(\cdot)$.

I tried to use consistent mathematical expressions throughout the thesis. For example, I use $\mathbb{R}$ for the set of real numbers, $\mathbf{X}$ for a data matrix, and $\mathbf{x}$ for the vector expression of a data item. However, because many mathematical expressions are used in the thesis, one character, e.g., $\mathbf{P}$ may refer to different matrices in different chapters (same for vectors, sets, etc.).

Chapter 2

# Background

This chapter first gives a brief introduction to clustering algorithms upon which the proposed semi-supervised clustering technique will be applied. It then introduces the dimension reduction technique which is often closely related to clustering. At last, it introduces clustering evaluation metrics that will be used throughout the thesis.

## 2.1 Overview of Clustering

Clustering techniques can be roughly categorized into two groups:

- Hierarchical methods
  - Agglomerative
  - Divisive
- Partitional methods

- Density-based
- Spectral
- Mixture-model-based

### 2.1.1 Hierarchical Clustering

Hierarchical clustering outputs a hierarchical structure of data items through a series of data fusions or partitions. These algorithms can be either *agglomerative* ("bottom-up") or divisive ("top-down") [JMF99]. Agglomerative algorithms treat each data item as a singleton cluster and merge them successively into larger clusters. Divisive algorithms treat the whole set of data items as a single cluster and continue to split it into successively smaller clusters until individual items are reached.

The hierarchical structure generated, which is also known as the *cluster dendrogram*, is informative for users to understand the data collection. Hierarchical clustering does not require the pre-knowledge of the number of clusters, which is often required by other types of clustering algorithms. Despite these advantages, hierarchical clustering is known for low efficiency. The most common hierarchical clustering algorithms have a complexity that is at least quadratic in the number of items to be clustered.

The crux of hierarchical clustering is measuring the distance between clusters. Depending on the choice of distance metrics, major agglomerative algorithms in-

clude single linkage, complete linkage, average linkage and average group linkage algorithms [MS99]. For recent work on divisive methods, Zhao et al. [ZHT05] use entropy as a measure of cluster inhomogeneity and greedily increase the size of partition by one in each iteration. Dubnov et al. [DEYG+02] recursively split clusters using a statistical transformation, and Boley [Bol98] proposes the principal direction divisive partitioning method.

### 2.1.2 Partitional Clustering

Unlike hierarchical clustering, the number of clusters is usually required in partitional clustering. Given the number of clusters $k$ known, partitional clustering methods divide data items into $k$ clusters in one step. Since no hierarchical structure is generated, partitional clustering algorithms are also known as *flat clustering*. Compared to hierarchical clustering, partitional clustering algorithms are computationally more efficient and can be used as the intermediate partition method in divisive hierarchical clustering. For example, the k-means and EM clustering method are partitional and both have linear complexity. Based on the clustering criterion adopted, partitional algorithms can be further categorized.

In density-based methods, clusters are viewed as regions in the data space in which the items are dense, and the decision boundary always lies in the low density region. Density-based algorithms can identify clusters of arbitrary shapes. However, because density-based algorithms apply a local clustering criterion, the

clustering solution is not globally optimal. DBSCAN [EKSX96] is a representative density-based clustering algorithm. Mean shift [CM02] is another representative density-based approach that detects the modes of density and assigns every data item to its corresponding density mode. Mean shift is widely used in computer vision applications, such as visual tracking and image segmentation.

Spectral clustering methods have emerged as one of the most effective clustering algorithms and have shown great success in many applications including computer vision, bioinformatics, speech recognition, VLSI design, and document clustering. By representing the proximities between data items into a graph format, spectral clustering involves finding the best cuts of the graph that optimizes certain predefined objective functions. Various objective functions have been proposed (e.g., average cut [CSZ93], average association [SM97a], normalized cut [SM97a], and min-max cut [DHZ$^+$01]). Directly optimizing graph cut objective functions is NP-hard. To find an approximate solution instead, a graph cut problem is transformed into an eigenvalue problem. Spectral clustering methods differ from other clustering methods in that they guarantee to find the global optima in terms of the predefined objective functions. Due to this reason, although an approximate solution, spectral clustering can often produce better result than direct optimization techniques such as k-means.

In mixture-model-based methods, each cluster is mathematically represented as a parametric distribution, for examples, Gaussian distribution for continuous

input values or Poisson distribution for discrete input values. Hence the entire data set is modeled by a mixture of these distributions. Well-studied statistical inference techniques are available to find parameters of the model. The EM algorithm [DLR77] [CS96] is a well-known technique for estimating the parameters in the general case. Besides, mixture-model-based methods enable "soft clustering", which means a data item is assigned to a cluster with a probability, as contrast to "hard clustering" where every item is assigned exclusively to one and only one cluster.

## 2.2 Representative Clustering Algorithms

### 2.2.1 k-means

k-means [Mac67a] [Llo82] is probably the most widely used clustering algorithm. k-means aims to partition $n$ observed data items into $k$ clusters where each item is assigned to the cluster with the nearest center. The objective of k-means is to minimize the sum of distance from every data point to its closest center. Let $\{\mathbf{c}_i\}|_{i=1}^{k}$ be the centroids of $k$ clusters. Let $\eta(\cdot)$ be the assignment function. Then $\eta(i) = j$ means the $i$th item is assigned to the $j$th cluster. k-means minimizes the following objective

$$\arg\min_{C,\eta} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{c}_{\eta(i)}\|^2 \tag{2.1}$$

Lowering the objective function leads to more compact clusters, where each item gets closer to its cluster centroid.

Finding the global optima for the k-means objective function is an NP-complete problem [GJW82]. Heuristic approximations have been proposed. *Lloyd's algorithm* is the most popular heuristics for solving k-means. It is based on a simple iterative scheme for finding a locally minimal solution [For65] [Mac67b]. The heuristic solution indicates that k-means can be viewed as a special case of EM that assumes

1. Each cluster is modeled by a spherical Gaussian distribution;

2. Each data item is assigned to one and only one cluster;

3. The mixture weights are equal.

The pseudocode for k-means is given in Algorithm 1.

---

**Algorithm 1:** k-means

---

**Input** : A set of data items $\{\mathbf{x}_i\}_{i=1}^n$;
the number of clusters $k$.
**Output**: A disjoint set of clusters.
1. Initialize clusters: cluster centroids $\{\mathbf{c}_j^{(0)}\}_{j=1}^k$ are chosen at random;
2. Repeat until converge:
   2a. Assign data items following the nearest neighbor rule, that is

$$\eta(i) = \arg \min_{j=1,\ldots,k} \|\mathbf{x}_i - \mathbf{c}_j^{(0)}\|^2$$

   2b. Update centroids

$$\mathbf{c}_j^{(t)} = \frac{\sum_{i:\eta(i)=j} \mathbf{x}_i}{n_j}$$

   2c. $t \leftarrow t + 1$

---

### 2.2.2 Spherical k-means (SPKM)

The spherical k-means algorithm (SPKM)[DHZ+01] is a k-means method that uses *cosine similarity* to measure the distance from every data point to its closest center. SPKM is a popular method for clustering high-dimensional text data. It is shown that the spherical k-means algorithm is one of the fastest document clustering algorithms [Zho05].

Before text data can be analyzed, the popular vector space model ("bag-of-words") [SWY97] is usually used to represent raw text data as high-dimensional vectors, where each dimension of the vector is a unique term [SM86]. The adoption of cosine similarity in SPKM is based on the observation that text vectors have only non-negative entries, and the high-dimensional text vectors have directional properties, i.e., the length of the vectors is much less discriminative than their direction. In SPKM, each data item as well as cluster centroids are normalized and are represented as unit-length vectors. The effect of this normalization is to only account for the direction of each vector but not the length, since the Euclidean distance between normalized vectors is equivalent to one minus the cosine similarity between the vectors.

### 2.2.3 Normalized Cut (NC)

Normalized Cut has been shown to be the one of the best spectral clustering approaches [SM97b]. In this thesis, I use Normalized Cut as a representative of

spectral clustering approaches to demonstrate idea and concept.

Spectral clustering approaches model a data set as an undirected graph, where each data item is a vertex in the graph, and the edge between two vertices $i, j$ is assigned a weight $w_{ij}$ to reflect the similarity between items $i$ and $j$. Let matrix $\mathbf{W}$ be the *affinity matrix* associated with the graph, such that $\mathbf{W}(i,j) = w_{ij}$. Let $C_i$, $C_j$ denote two clusters of the given data set $S$, and $\mathbf{W}(C_i, C_j)$ denote the sum of similarities between the two clusters $C_i$ and $C_j$

$$\mathbf{W}(C_i, C_j) = \sum_{u \in C_i, v \in C_j} w_{uv} \tag{2.2}$$

The objective of Normalized Cut is to minimize

$$\sum_{i=1}^{k} \frac{\mathbf{W}(C_i, \overline{C}_i)}{\mathbf{W}(C_i, S)} \tag{2.3}$$

where $k$ is the number of clusters. The numerator $\mathbf{W}(C_i, \overline{C}_i)$ measures how tightly the cluster $C_i$ is connected to the rest of the data set, while the denominator measures how compact the entire data set is. To find the approximate solution, let $\mathbf{x}_i = (x_{1i}, \ldots, x_{ni})^T$ be the indicator vector of the cluster $C_i$ in which each element $x_{ki}$ takes a binary value $\{0, 1\}$ to indicate if the $k$'th item in data set belongs to $C_i$ or not. After introducing indicator vectors, the objective function of NC is

$$k - \sum_{i=1}^{k} \mathbf{y}_i^T \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} \mathbf{y}_i \tag{2.4}$$

where $\mathbf{D}$ is the diagonal row sum matrix, i.e., $\mathbf{D}_{ii} = \sum_l \mathbf{W}_{li}$, $\mathbf{y}_i = \frac{\mathbf{D}^{1/2}\mathbf{x}_i}{\|\mathbf{D}^{1/2}\mathbf{x}_i\|}$, and $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix having the same order than $\mathbf{Y}^T\mathbf{Y}$. As introduced in Section 2.1.2, directly optimizing equation 2.4 is NP-hard. If we relax the problem by letting the indicator vectors $\mathbf{x}_i$ take real values, the optimization problem can be easily solved under the constraint $\mathbf{Y}^T\mathbf{Y} = \mathbf{I}$. As shown by Golub et al. [GL89], when $\mathbf{y}_1, \ldots, \mathbf{y}_k$ are $k$ eigenvectors associated with the $k$ largest eigenvalues of matrix $\mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$, the Normalized Cut objective function reaches the minimum, and $\mathbf{y}_1, \ldots, \mathbf{y}_k$ encode the cluster membership information of the given data set. However, since these eigenvectors take real values for their elements, they do not directly indicate the cluster membership for each data item. A common approach for deriving the final cluster set is to project each data point into the eigenspace spanned by the above $k$ eigenvectors, and apply the k-mean algorithm within this eigen-space.

## 2.3 Overview of Feature Reduction with Transformations

Feature reduction is closely related to clustering. Both techniques focus on the intrinsic structure of a data set. Feature reduction aims to preserve the structure with fewer features, while clustering aims to identify the structure for the purpose of data analysis. Because many clustering approaches lose algorithmic effectiveness

when handling high-dimensional data, a feature reduction step is usually adopted to enhance clustering performance.

Feature reduction can be achieved by linear or non-linear transformations. Two classical approaches of finding optimal linear transformations are Principal Component Analysis (PCA) [Hot33] for the unsupervised setting, and Linear Discriminant Analysis (LDA) for the supervised setting. The linear transformation methods can be less efficient when sever non-linearity is involved in the data. To this end, Kernel PCA [SSM97] and Kernel LDA [MRW+99] are developed using the popular kernel technique [STC04]. Besides, multidimensional scaling (MDS) is another widely used linear feature reduction method. PCA, LDA, and MDS are all *global methods* in that they preserve only the global structure of a data set. To overcome the drawbacks of global methods and their variants, a number of *local* dimension reduction methods have been proposed, such as Laplacian Eigenmaps [BN02], Locally Linear Embedding (LLE) [RS00] and Locality Preserving Projections (LPP) [HN03]. Local methods embed data in the low-dimensional space such that nearby data points in the original space are still near to each other in the embedded space. Local methods are particularly useful for data whose local geometry is close to Euclidean, but whose global geometry may not be.

Compared to nonlinear methods, linear transformations of features are particularly attractive because they are simple to compute and are analytically tractable. For the above reasons, in this thesis, I focused on linear transformations.

### 2.3.1 Principal Component Analysis (PCA)

PCA is an unsupervised dimension reduction technique. PCA seeks a projection that best represents the data in the reduced space. The optimal projection is defined in the least squares sense. Assume that we have $n$ $f$-dimensional data vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, which form the $f \times n$ data matrix $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \ldots \ \mathbf{x}_n]$. The matrix $\mathbf{X}$ is decomposed into $\mathbf{X} \approx \mathbf{SA}$, where $\mathbf{S}$ is a $f \times r$ matrix, $\mathbf{A}$ is a $r \times n$ matrix and $r \leq f$. The reconstruction error is defined as

$$\mathcal{E} = \|\mathbf{X} - \mathbf{SA}\|_F^2 = \sum_{i=1}^{f} \sum_{j=1}^{n} (x_{ij} - \sum_{k=1}^{r} s_{ik} a_{kj})^2 \tag{2.5}$$

and the PCA projection minimizes $\mathcal{E}$. It can be showed that $\mathcal{E}$ is minimized when the column vectors of $\mathbf{S}$, which are $\{\mathbf{s}_i\}|_{i=1}^{r}$, are the $r$ eigenvectors of the scatter matrix $\mathbf{H} = \sum_{i=1}^{n} (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T$ corresponding to the largest eigenvalues. Then $\mathbf{A}$ is the optimal rank $r$ approximation of $\mathbf{X}$. Before applying PCA reduction, the data should be mean-removed. This preprocessing ensures that the matrix $\mathbf{S}$ will not be affected by the location of the center of the data. With mean removed, the scatter matrix $\mathbf{H}$ is in fact a covariance matrix. Therefore, PCA preserves the variance of data. Oftentimes the covariance matrix only has a few large eigenvalues. This implies that the $r$-dimensional subspace contains the signal and the remaining $f - r$ dimensions generally contain noise.

### 2.3.2 Linear Discriminant Analysis (LDA)

Contrary to PCA, LDA is a supervised dimension reduction technique. Besides a set of data vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, LDA also requires the class assignment of each vector as prior knowledge. Whereas PCA seeks directions that are efficient for representation, i.e. maximally preserves variances, discriminant analysis seeks directions that are efficient for discrimination.

To that purpose LDA maximizes the following objective

$$J(\mathbf{a}) = \frac{\mathbf{a}^T \mathbf{S}_B \mathbf{a}}{\mathbf{a}^T \mathbf{S}_W \mathbf{a}} \tag{2.6}$$

where $\mathbf{a}$ is a projection vector, $S_B$ is the "between classes scatter matrix" and $\mathbf{S}_W$ is the pooled "within classes scatter matrix" defined as

$$\mathbf{S}_B = \sum_C (\mu_C - \bar{\mathbf{x}})(\mu_C - \bar{\mathbf{x}})^T \tag{2.7}$$

$$\mathbf{S}_W = \sum_C \sum_{i \in C} (\mathbf{x}_i - \mu_C)(\mathbf{x}_i - \mu_C)^T \tag{2.8}$$

where $\bar{\mathbf{x}}$ is the overall mean of the data set, and $\mu_C$ is the mean of class $C$.

The objective says that the optimal LDA solution is the one where classes are well separated, measured in terms of class-means, and each class is compact, measured in terms of pooled variances of the data items assigned to a particular class.

### 2.3.3 Locality Preserving Projections (LPP)

LPP is a linear dimension reduction technique that is applicable to both unsupervised and supervised settings. Unlike PCA and LDA, LPP is a local method that optimally preserves local neighborhood information of a data set.

LPP first constructs the adjacency graph of data, where each data item is a vertex in the graph. Let $\mathbf{W}$ be the adjacency matrix associated with the graph. Then a non-zero $w_{ij}$ stands for the existence of an edge between items $i$ and $j$ if item $i$ is near to item $j$, i.e., $j$ is in the $k$ nearest neighbors of $i$ for some $k$. Let $\mathbf{y}_i$ and $\mathbf{y}_j$ be the projections of $\mathbf{x}_i$ and $\mathbf{x}_j$ in the $r$-dimensional space. LPP minimizes the following objective function

$$J(\mathbf{a}) = \sum_{i,j}(\mathbf{y}_i - \mathbf{y}_j)^2\mathbf{W}_{i,j} \tag{2.9}$$

$$= \sum_{i,j}(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j)^2\mathbf{W}_{i,j} \tag{2.10}$$

under the constraints

$$\mathbf{a}^T\mathbf{X}\mathbf{D}\mathbf{X}^T\mathbf{a} = 1 \tag{2.11}$$

where $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, and $\mathbf{D}$ is the row (or column) sum matrix where $\mathbf{D}_{ii} = \sum_l \mathbf{W}_{li}$. Minimizing Equation 2.10 is an attempt to ensure and if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close then their projections in the $r$-dimensional space are close as well.

## 2.4 Evaluation of Clustering

The quality of clustering can be evaluated by either an *internal criterion* or an *external criterion*. An internal criterion is usually adopted in constructing the objective function of a clustering algorithm, formalizing the goal of achieving high intra-cluster similarity (i.e., items within a cluster are similar) and low inter-cluster similarity (i.e., items from different clusters are dissimilar). For example, in k-means clustering, the value of the objective function when the algorithm converge indicates the quality of the data partition. But good score on an internal criterion does not necessarily leads to good effectiveness in an application. Moreover, different clustering approaches adopt different objective functions. The objective function scores by different approaches are not comparable to indicate which clustering approach generates better quality clusters. Alternatively, we can apply clustering approaches to an evaluation benchmark or gold standard. For example, labeled data can be used in evaluation. We remove the class labels and apply a clustering algorithm to the data set, then evaluate how well the clustering solution matches the class labels by using an external criterion.

There are four widely adopted external criteria of clustering quality

1. *Purity*

2. *Normalized Mutual Information* (NMI)

3. *Rand Index* (RI)

4. *F-measure* (F)

To calculate Purity, each cluster is assigned to the class which is most frequent in the cluster, and Purity is the accuracy of this assignment which is evaluated by counting the number of correctly assigned items divided by the total number of items. However, Purity is biased towards large number of clusters. In particular, Purity is 1 if each data item forms its own cluster. The three other metrics are not influenced by the number of clusters and are adopted in this thesis.

Given the true class labeling of a data set, Normalized Mutual Information (NMI) measures how closely a clustering algorithm can reconstruct the true label distribution of the data [SSGM00]. Let $C$ and $\hat{C}$ be the random variables denoting the data partitions based on the ground truth and a clustering algorithm, respectively. Then NMI is defined as

$$NMI(C, \hat{C}) = \frac{2I(C; \hat{C})}{H(C) + H(\hat{C})} \tag{2.12}$$

where $I(C; \hat{C}) = H(C) - H(C|\hat{C})$ is the mutual information between the random variables $C$ and $\hat{C}$. $H(C)$ is the Shannon entropy of $C$, and $H(C|\hat{C})$ is the conditional entropy. NMI is a preferred and widely used metric because it does not suffer from biases like purity, entropy, and the F-measure. Singletons are not evaluated as perfect.

The Rand Index (RI) measures the degree of similarity in terms of pairwise

co-assignments between the cluster membership $C$ from the ground truth and the solution $\hat{C}$ generated by a clustering algorithm. It is defined as

$$RI(C, \hat{C}) = \frac{|c_i = c_j \wedge \hat{c}_i = \hat{c}_j| + |c_i \neq c_j \wedge \hat{c}_i \neq \hat{c}_j|}{n(n-1)/2} \tag{2.13}$$

where $c_i$ and $\hat{c}_i$ are the cluster membership of item $i$ according to $C$ and $\hat{C}$, and $n$ is the number of data items being clustered. Obviously, RI penalizes both the false positive and false negative decisions during clustering.

It is possible to penalize each type of error with different weight and this is achieved by the F-measure [vR79]. Let $\mathbb{T}$ denote the set of pairs of data items that belong to a same cluster according to ground truth and $\mathbb{R}$ denote the set of pairs of data items that have been assigned to a same cluster by the clustering algorithm. Then, *precision* and *recall* are defined as

$$\begin{aligned} precision &= \frac{|\mathbb{R} \cap \mathbb{T}|}{|\mathbb{R}|} \\ recall &= \frac{|\mathbb{R} \cap \mathbb{T}|}{|\mathbb{T}|} \end{aligned}$$

And the general form of F-measure is defined as

$$F_\beta = \frac{(1+\beta^2)(precision \times recall)}{\beta^2 \times precision + recall} \tag{2.14}$$

When $\beta = 1$, precision and recall are evenly weighted, and it is the commonly used

F-score, which is also known as the $F_1$ measure, defined as

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \tag{2.15}$$

The $F_1$ score is used in the thesis for clustering evaluation.

Chapter 3

# Related Work

This chapter briefly reviews existing work on semi-supervised learning (SSL), with focus on semi-supervised clustering approaches.

## 3.1 Semi-Supervised Classification and Semi-Supervised Regression

Semi-supervised classification and regression try to explore the wealth of unlabeled data to improve the accuracy of a learner. Thus, the training data for semi-supervised classification and regression tasks include labeled data plus unlabeled data. Usually, the amount of labeled data is small, but unlabeled data are available in large amount.

Semi-supervised classification and regression approaches are based on three major assumptions

- Cluster assumption says that if items are in the same cluster, they are likely to belong to the same class.
- Low density assumption says that the decision boundary always lie in a low density region.
- Manifold assumption says that if two items $\mathbf{x}_1$, $\mathbf{x}_2$ are linked by a path of high density then their outputs $\mathbf{y}_1$, $\mathbf{y}_2$ are likely to be close.

To incorporate the semi-supervised assumptions in the classification and regression objectives, usually an additional regularization term on the distribution or geometry of both labeled and unlabeled samples is introduced. Transductive Support Vector Machines (TSVM)[Vap98], Laplacian SVM (LapSVM)[BNS06], Laplacian Regularized Least Squares (LapRLS)[BNS06], semi-supervised mixture models, and semi-supervised entropy minimization [GB04] are some representative works. Besides, Zhu et al. [ZGL03] exploits the manifold structure of labeled and unlabeled samples by Gaussian fields and harmonic functions. Moreover, self-training [Yar95] and co-training [NG00] represent another major branch of semi-supervised classification and regression approaches.

## 3.2 Semi-Supervised Clustering

Semi-supervised clustering tries to cluster data items with the help of side information. Side information can take various forms, such as restrictions on the number

of data items in a cluster and the variance of a cluster [GEJD07], a few labeled data items [BBM02], or pairwise constraints. Side information in the form of pairwise constraints is most general. I study pairwise constraints in this thesis and briefly review semi-supervised clustering approaches using pairwise constraints in this section.

According to how constraints are explored, semi-supervised clustering techniques fall into three major categories, which are constraint enforcement, distance metric learning, and dimension reduction.

### 3.2.1 Constraint Enforcement

One can enforce constraints during the clustering process. For example, Wagstaff et al. [WC00] [WCRS01] propose to adjust the data item assignment step in k-means, such that none of the constraint is violated in the final k-means solution. This clustering approach is known as constrained k-means. Because every pair of constraint should be strictly conformed, this kind of methods may encounter the over-constrained problem where no solution can be found. For example, Davidson et al. [DR05] study the feasibility issue of k-means clustering for each type of pairwise constraints. Another way to enforce side information in k-means is to initialize k-means with labeled data items [BBM02]. Instead of heuristically modifying the cluster assignment step in k-means, Basu et al. [BBM04a] propose to model the constrained k-means problem based on Hidden Markov Random Fields.

Moreover, Blum et al [BLRR04] and Ji et al. [JX06] propose frameworks to explore pairwise constraints for spectral clustering methods.

### 3.2.2 Distance Metric Learning

One can also learn a distance metric based on constraints, and use the learned metric to measure pairwise distance between items in clustering. Learning distance metric is equivalent to learning an adaptive feature weighting scheme. Note that all the clustering techniques rely on some notion of pairwise distance between data items. Instead of assigning the same weight to every feature, the relative importance of features can be learned from constraints. Thus, the learned distance metric evaluates pairwise distances between items better. Xing et al. [XNJR03] consider a distance metric in the form of general Mahalanobis distance and use convex optimization and iterative projections to learn it. Cohn et al. [CH00] learns the distance metric in a probabilistic setting for EM clustering, which is equivalent to learning the Kullback-Leibler divergence. Klein et al. [KKM02] use all pairs shortest path algorithm to adjust the squared Euclidean distance. The effectiveness of metric learning can be improved through boosting. Hertz et al. [HBhW04] learns the distance by boosting weak learners based on partitioning the original feature space. [LJJ07] do metric learning iteratively to take full advantage of the sparse constraints.

One problem with distance metric learning is that the learning is not efficient

nor effective for high-dimensional data. Distance learning is usually reduced to solving a convex optimization problem with gradient descent and iterative projection, and often suffers from large computation cost. The number of parameters to be learned equals to or is quadratic to the feature space dimensions. Besides, it has been shown that some metric learning methods may even degrade the clustering performance if applied to the high dimensional sparse feature spaces, although they work pretty well with low dimensional data [TXZW07].

### 3.2.3 Dimension Reduction

One can use constraints to define an optimal subspace, such that data represented in the subspace show a more evident clustering structure or data are distributed in a way that conforms to pairwise constraints.

Bar-Hillel et al. [BHHSW03] propose Relevant Component Analysis (RCA) that changes the feature space via a global linear transformation where relevant features are assigned with larger weights. RCA can only handle pairwise must-link constraints. Hoi et al. [HLLM06] extends RCA and propose Discriminative Component Analysis (DCA) that can use both must-link and cannot-link constraints. An et al. [ALV08] propose to incorporate constraints using a modified Locality Preserving Projections (LPP) cost function. All the above dimension-reduction-based approaches explore constraints only and do not consider the usefulness of abundant unconstrained data. With sparse constraints, the methods face the over-

fitting problem. That is, the subspace that best satisfies a few pairs of constraints does not necessarily reveal the true structure of the entire data set. To this end, Zhang et al. [ZZC07] and Cevikalp et al. [CVJK08] propose semi-supervised dimension reduction methods that explore both constraints and unconstrained data. However, both methods require users to intuitively set parameters to balance the constrained and the unconstrained data. The Dual Subspace Projections (DSP) approach introduced in Chapter 6 does not overfit and is able to explore constraints and unconstrained data in a principled way.

Moreover, Tang et al. [TXZW07] propose to place data items into groups based on constraints and then maximally separate the data groups. The clustering method proposed in Chapter 4 based on Approximate Structure Preserving (ASP) dimension reduction shares similar idea, but is computationally more efficient and has more robust performance. Yan et al. [YD06] propose to project data and constraints in multiple subspaces, where metric learning and clustering are performed. Then the ensemble clustering result is the final clustering solution. This method divides the metric learning problem for high-dimensional data into many metric learning problems for low-dimensional data and aggregates the results. However, the computation cost is still high, and the clustering performance is highly dependent on the ensemble approach adopted. Yan et al. [YWLG09] propose to first perform semi-supervised dimension reduction, then do distance learning in the reduced space. Oncel et al. [TPM09] propose a semi-supervised kernel mean

shift method that explore must-link constraints in the kernel space. The method is effective, but is unable to use cannot-link information.

Chapter 4

# Semi-Supervised Clustering by Approximate-Structure-Preserving Dimension Reduction

## 4.1 Motivation

Pairwise constraints define an "approximate-clustering structure on a data set". For example, suppose data items $\mathbf{x}_i$ and $\mathbf{x}_j$ are "must-linked" while $\mathbf{x}_p$ and $\mathbf{x}_q$ are "cannot-linked" according to pairwise constraints. Even though the cluster membership of each data item is unknown, in order to satisfy constraints, data items $\mathbf{x}_i$ and $\mathbf{x}_j$ should be close to each other, while $\mathbf{x}_p$ and $\mathbf{x}_q$ should be far apart from each other. The pairwise cluster-membership relations among $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_p$, and $\mathbf{x}_q$ "approximately" outline the desired clustering structure of the entire data set.

To explore the approximate-clustering structure defined by constraints, a clustering problem can be formulated as a semi-supervised clustering task by *approximate-structure-preserving (ASP)* dimension reduction. That is, we seek a projection $\mathbf{U} \in \mathbb{R}^{f \times r}$ to project input data $\mathbf{X} \in \mathbb{R}^{f \times n}$ onto a much reduced-dimension subspace by

$$\widehat{\mathbf{X}} = \mathbf{U}^T \mathbf{X} \tag{4.1}$$

where $\widehat{\mathbf{X}} \in \mathbb{R}^{r \times n}$ is data represented in the $r$-dimensional reduced space, $r \ll f$. The approximate-clustering structure defined by pairwise constraints is more clear in the reduced space $\widehat{\mathbf{X}}$ than in the full space $\mathbf{X}$. This purpose is achieved by splitting the $f$-dimensional input space into an $r$-dimensional space which contains all the *structure-relevant* dimensions (i.e., attributes), and an $s$-dimensional space $(r + s = f)$ which contains all the *structure-irrelevant* dimensions (i.e., noise). Suppose $\mathbf{x}$ is a data vector in the full space and $\mathbf{P}$ is some projection matrix, then, we can split the space into $\mathbf{P}^T\mathbf{x} = [\mathbf{U}_r\ \mathbf{V}_s]^T\mathbf{x}$, or more explicitly

$$\begin{bmatrix} \widehat{\mathbf{X}} \\ \widehat{\mathbf{X}}^{\perp} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_r^T\mathbf{X} \\ \mathbf{V}_s^T\mathbf{X} \end{bmatrix} \tag{4.2}$$

where $\widehat{\mathbf{X}}$ is in $r$-dimensional relevant space, and $\widehat{\mathbf{X}}^{\perp}$ is in $s$-dimensional subspace of noise orthogonal to the relevant space. A desired projection matrix $\mathbf{P} = [\mathbf{U}_r\ \mathbf{V}_s]$ should satisfy

$$\widehat{\mathbf{X}}^{\perp} = \mathbf{V}_s^T\mathbf{X} = 0 \tag{4.3}$$

which means that the structure-irrelevant noise that exists in the full space is removed, and only relevant dimensions are kept in the reduced space. Thus, the subspace representation of data is $\widehat{\mathbf{X}} = \mathbf{U}_r^T\mathbf{X}$. To achieve this subspace representation, all we need is to find the projection $\mathbf{U}$ (i.e., no need to find $\mathbf{V}$).

The ASP method is based on this motivation and can effectively preserve the approximate-clustering structure in the reduced space. The projection $\mathbf{U}$ can be generated by well-studied matrix factorization techniques. Thus, ASP is efficient for dealing with large data sets that are usually encountered in real applications.

## 4.2 Outline

This section presents a step-by-step description to the ASP dimension reduction method as well as the corresponding semi-supervised clustering method. Theoretical justification is given in the next section.

Given a collection of pairwise constraints, the first step is to do transitive closure to all the data items involved in must-link constraints since a must-link constraint is a binary equivalence relation. For example, if items **a** and **b** are must-linked, **b** and **c** are must-linked, then items **a** and **c** are must-linked too. Note that, performing transitive closure is a common preprocessing step in semi-supervised learning. The assumption is that constraints are error free. The assumption holds well if constraints are generated by human efforts but does not hold well if constraints are derived from domain knowledge by some automatic schemes. Chapter 5 will

discuss this problem more and introduces a method that performs robustly even with noisy constraints.

After performing transitive closure, a set of data items are partitioned into several *data groups*. Given $n$ $f$-dimensional data vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n$, which form the data matrix $\mathbf{X} \in \mathbb{R}^{f \times n}$, $f \gg n$, suppose $b$ data groups are generated. For any given data group $B_i$, $i = 1, \cdots, b$, suppose the group contains $m$ data items (i.e., $|B_i| = m$), and let matrix $\mathbf{X}_i = [\mathbf{x}_{i1} \ \cdots \ \mathbf{x}_{im}] \in \mathbb{R}^{f \times m}$ represent the collection of the $m$ data items. The next step is to calculate the centroid $\mathbf{c}_i$ of group $B_i$ as $\mathbf{c}_i = \frac{1}{m} \sum_m \mathbf{x}_{im}$. The centroid $\mathbf{c}_i \in \mathbb{R}^f$ is simply a rank-1 approximation to the data group $B_i$.

The relations between data groups are either cannot-linked or do-not-know, while the relations between the data items within a group are must-linked. Thus, data groups reflect the approximate-clustering structure defined by constraints. Such approximate-clustering structure can be encoded into a *representative matrix* as follows

**Definition 1.** ***Representative Matrix*** *Let matrix* $\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_b \end{bmatrix} \in \mathbb{R}^{f \times b}$ *be a* representative matrix*, where the* $i^{th}$ *column,* $\mathbf{c}_i$*, is the centroid vector of data group* $B_i$*.*

Given a representative matrix $\mathbf{C} \in \mathbb{R}^{f \times b}$, the *orthonormal basis* $\mathbf{U}$ for $range(\mathbf{C})$ (from Definition 2) is the desired projection. Suppose $rank(\mathbf{C}) = r$, where $rank(\cdot)$ denotes matrix rank, then $\mathbf{U} \in \mathbb{R}^{f \times r}$. Note that, for data whose feature space

dimension is much bigger than the number of data items, i.e., $r < n \ll f$, data can be directly projected to the $r$-dimensional space. In this case, the dimension of the reduced space is automatically determined by the rank of matrix $\mathbf{C}$. Given the projection $\mathbf{U}$, input high-dimensional data are projected onto a reduced space using Equation 4.1. After the projection, data are in the $r$-dimensional space ($r \ll f$). The ordinary k-means clustering method is applied to the reduced space to generate data partitions.

For easier reference, the definition of *range* is given as follows

**Definition 2.** *The range of matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *is the set of all vectors* $\mathbf{y} \in \mathbb{R}^m$ *that can be expressed as* $\mathbf{Ax}$ *for some* $\mathbf{x} \in \mathbf{R}^n$*:* $range(\mathbf{A}) = \{\mathbf{y} | \mathbf{y} = A\mathbf{x},\ \mathbf{x} \in \mathbf{R}^n\}$.

The main steps of semi-supervised clustering by approximate-structure-preserving dimension reduction are summarized in Algorithm 2.

**Algorithm 2:** ASP

**Input** : A set of data items $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$, $\mathbf{X} \in \mathbb{R}^{f \times n}$ ($f \gg n$);
A set of must-link and cannot-link constraints $\Omega_M = \{(\mathbf{x}_i, \mathbf{x'}_i)\}$ and $\Omega_C = \{(\mathbf{x}_i, \mathbf{x'}_i)\}$;
The number of desired clusters $k$.

**Output**: A disjoint set of clusters

1 Do transitive closure to generate $b$ data groups;
2 Compute centroids $\mathbf{c}_i \in \mathbb{R}^f$ for each data group $B_i$, $i = 1, \cdots, b$;
3 Compose the representative matrix $\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_b \end{bmatrix} \in \mathbb{R}^{f \times b}$, where each column $\mathbf{c}_i$ is a centroid vector;
4 Find the orthonormal basis $\mathbf{U} \in \mathbb{R}^{f \times r}$ for $range(\mathbf{C})$, where $rank(\mathbf{C}) = r$;
5 Project data using $\mathbf{U}$. That is $\widehat{\mathbf{X}} = \mathbf{U}^T\mathbf{X}$, and $\widehat{\mathbf{X}} \in \mathbb{R}^{r \times n}$ is the reduced $r$-dimensional representation of data;
6 Apply k-means clustering method to $\widehat{\mathbf{X}}$ to generate data partition that better meets the clustering requirements defined by pairwise constraints.

## 4.3 Analysis and Validation

In this section, the correctness of the ASP method is validated. To measure the clustering structure of a data set, two concepts, *overall volume* and *group volume* are defined. Overall volume $Vol$ measures the spread of the entire collection of data items, and group volume $Vol_i$ measures the spread of data items that belong to the $i$th data group.

**Definition 3** (Overall Volume)**.** *Given a collection of data items distributed into $b$ groups $B_i, i = 1, \ldots, b$, and $c_i$ be the centroid of group $B_i$, let $\mathbf{c}$ be the overall centroid of the entire data collection. Then the overall volume $Vol$ is defined as* $Vol = \frac{1}{b}\sum_{i=1}^{b} \|\mathbf{c}_i - \mathbf{c}\|_2^2$.

**Definition 4** (Group Volume)**.** *Given a data group $B_i$ that contains $n_i$ data items $\mathbf{x}_{ij}$, $j = 1, \cdots, n_i$, the volume $Vol_i$ of group $B_i$ is defined as* $Vol_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \|\mathbf{x}_{ij} - \mathbf{c}_i\|_2^2$ *where* $\mathbf{c}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathbf{x}_{ij}$ *is the centroid of group $B_i$.*

The ASP method has the properties that, after dimension reduction, the overall volume is strictly preserved, while the volume of a data group is decreased, as stated below.

**Property 1** (Constant Overall Volume)**.** *The overall volume $Vol$ is strictly preserved in the subspace: $\widehat{Vol} = Vol$, where $\widehat{Vol}$ denotes the overall volume after ASP dimension reduction.*

**Property 2** (Group Volume Shrinkage)**.** *The volume of any given data group $B_i$ shrinks in the subspace: $\widehat{Vol_i} < Vol_i$ , where $\widehat{Vol_i}$ denotes the volume of the ith group after ASP dimension reduction.*

The following metric is defined to measure the clustering structure of a data set

$$Vol_{diff} = \frac{\sum_{i=1}^{b} Vol_i}{Vol} \tag{4.4}$$

Given a fixed overall volume $Vol$, smaller $Vol_{diff}$ means that data items from the same group are more packed together, and thus the clustering structure of the data collection is more evident. According to Properties 1 and 2, the following inequation holds.

$$\frac{\sum_{i=1}^{b} \widehat{Vol_i}}{\widehat{Vol}} < \frac{\sum_{i=1}^{b} Vol_i}{Vol} \tag{4.5}$$

Inequation 4.5 shows that the data group structure, which represents the approximate-clustering structured defined by constraints, is preserved and is more evident in the reduced space. Note that, both "must-link" and "cannot-link" constraints are satisfied in the ASP method. Property 2 guarantees that if two data items are must-linked, they will get closer to each other in the subspace, while Property 1 keeps two cannot-linked data items apart by preserving the constant center-to-center distance for the two data groups where the two cannot-linked data items belong to.

To prove the properties, I first restate the definition of *left null space* here for

easier reference, and then show Lemma 1 that will be used in the proofs.

**Definition 5** (Left Null Space)**.** *The left null space of matrix* $\mathbf{A} \in \mathbb{R}^{m \times n}$ *is the set of all vectors* $\mathbf{y} \in \mathbb{R}^m$ *such that* $\mathbf{A}^T\mathbf{y} = 0$*:* $lnull(\mathbf{A}) = \{\mathbf{y} | \mathbf{A}^T\mathbf{y} = 0\}$.

**Lemma 1** (Orthogonal Matrix Composition)**.** *Let* $\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_r] \in \mathbb{R}^{f \times r}$ *be an orthonormal basis of* $range(\mathbf{C})$*, and* $\mathbf{V} = [\mathbf{v}_1 \cdots \mathbf{v}_s] \in \mathbb{R}^{f \times s}$ *be an orthonormal basis of* $lnull(\mathbf{C})$*. Then* $\mathbf{P} = \begin{bmatrix} \mathbf{U} & \mathbf{V} \end{bmatrix} = [\mathbf{u}_1 \cdots \mathbf{u}_r \ \mathbf{v}_1 \cdots \mathbf{v}_s]$ *is an orthogonal matrix, and its columns form an orthonormal basis of* $\mathbb{R}^f$.

Lemma 1 can be easily proved based on the fact that $lnull(\mathbf{C})$ is the orthogonal complement of $range(\mathbf{C})$, that is

$$\mathbb{R}^f = \mathbf{U} \oplus \mathbf{V}, \ and \ r + s = f \tag{4.6}$$

Now, let us prove Properties 1 and 2.

**Proof 1** (Property 1)**.** *Let* $\mathbf{V}$ *be an orthonormal basis of* $lnull(\mathbf{C})$*. According Definition 5,* $\mathbf{V}^T\mathbf{C} = 0$ *. Let* $\mathbf{U}$ *be an orthonormal basis of the of* $range(\mathbf{C})$*. According to Lemma 1,* $P = \begin{bmatrix} \mathbf{U} & \mathbf{V} \end{bmatrix}$ *is an orthogonal matrix. Moreover, for data group* $B_i$*, we have* $\widehat{\mathbf{c}}_i = \frac{1}{n_i}\sum_j \widehat{\mathbf{x}}_{ij} = \frac{1}{n_i}\sum_j \mathbf{U}^T\mathbf{x}_{ij} = \mathbf{U}^T\mathbf{c}_i$*. According to the norm preserving property of an orthogonal transformation, we have*

$$\|\mathbf{c}_i - \mathbf{c}\|_2^2 \tag{4.7}$$

$$= \|\mathbf{P}^T(\mathbf{c}_i - \mathbf{c})\|_2^2 \tag{4.8}$$

$$= \|\mathbf{U}^T(\mathbf{c}_i - \mathbf{c})\|_2^2 + \|\mathbf{V}^T(\mathbf{c}_i - \mathbf{c})\|_2^2 \quad (4.9)$$

$$= \|\widehat{\mathbf{c}}_i - \widehat{\mathbf{c}}\|_2^2 + \|\mathbf{V}^T(\mathbf{c}_i - \mathbf{c})\|_2^2 \quad (4.10)$$

$$= \|\widehat{\mathbf{c}}_i - \widehat{\mathbf{c}}\|_2^2 \quad (4.11)$$

*where* $\|\mathbf{V}^T(\mathbf{c}_i - \mathbf{c})\|_2^2 = 0$ *follows the fact that* $\mathbf{V}^T\mathbf{C} = 0$*, and centroid* $\mathbf{c}$ *is a linear combination of group centroids* $\mathbf{c}_i$*:* $\mathbf{c} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i = \frac{1}{n}\sum_{i=1}^{b} n_i\mathbf{c}_i$*. Thus,* $\widehat{Vol} = \frac{1}{b}\sum_{i=1}^{b}\|\widehat{\mathbf{c}}_i - \widehat{\mathbf{c}}\|^2 = Vol = \frac{1}{b}\sum_{i=1}^{b}\|\mathbf{c}_i - \mathbf{c}\|^2$

**Proof 2** (Property 2)**.** *We know* $\mathbf{V}^T\mathbf{C} = 0$

$$\|\mathbf{x}_i - \mathbf{c}_i\|_2^2 \quad (4.12)$$

$$= \|\mathbf{P}^T(\mathbf{x}_i - \mathbf{c}_i)\|_2^2 \quad (4.13)$$

$$= \|\mathbf{U}^T(\mathbf{x}_i - \mathbf{c}_i)\|_2^2 + \|\mathbf{V}^T(\mathbf{x}_i - \mathbf{c}_i)\|_2^2 \quad (4.14)$$

$$= \|\widehat{\mathbf{x}}_i - \widehat{\mathbf{c}}_i\|_2^2 + \|\mathbf{V}^T\mathbf{x}_i\|_2^2 \quad (4.15)$$

$$> \|\widehat{\mathbf{x}}_i - \widehat{\mathbf{c}}_i\|_2^2 \quad (4.16)$$

*Therefore* $\widehat{Vol}_i = \frac{1}{m}\sum_{j=1}^{m}\|\widehat{\mathbf{x}_{ij}} - \widehat{\mathbf{c}}_i\|_2^2 \leq Vol_i = \frac{1}{m}\sum_{j=1}^{m}\|\mathbf{x}_{ij} - \mathbf{c}_i\|_2^2$.

## 4.4 Structure-Irrelevant Noise Deduction

This section introduces how the ASP method can split a feature space and remove the structure-irrelevant portion in the reduced space. Since $\mathbb{R}^f = \mathbf{U} \oplus \mathbf{V}$ (Lemma 1), every data item $\mathbf{x} \in \mathbb{R}^f$ in the full-dimension space can be equally expressed in

the form

$$\mathbf{x} = \mathbf{u} + \mathbf{v} \tag{4.17}$$

with $\mathbf{u}$ in $\mathbf{U} \in \mathbb{R}^r$ and $\mathbf{v}$ in $\mathbf{V} \in \mathbb{R}^s$, where $r+s = f$. $\mathbf{u}$ is the portion of $\mathbf{x}$ that is in the range of the representative matrix $\mathbf{C}$. That is, this portion of data item $\mathbf{x}$ can be linearly represented by the centroids of data groups. Since the representative matrix encodes the approximate-clustering structure, the portion $\mathbf{u}$ is structure-relevant. On the other hand, $\mathbf{v}$ is orthogonal to $range(\mathbf{C})$, which means that the $\mathbf{v}$ portion is structure-irrelevant. By ASP, the reduced-dimension representation of a data item is

$$\widehat{\mathbf{x}} = \mathbf{U}^T\mathbf{x} = \mathbf{U}^T(\mathbf{u} + \mathbf{v}) = \mathbf{U}^T\mathbf{u} \tag{4.18}$$

This implies that a feature space is split and only the structure-relevant portion is preserved by the ASP method.

The space-splitting view is useful to quantitatively measure how effective the ASP method preserves and strengthens the approximate-clustering structure in the reduced space. We have proved that the volume of a data group shrinks in the reduced space through ASP reduction (Property 2). The amount of shrinkage can be explicitly measured. For example, for a data group $B_i$ with $n_i$ data items, the volume of $B_i$ in the full space is

$$Vol_i \;=\; \frac{1}{n_i}\sum_{j=1}^{n_i} \|\mathbf{x}_{ij} - \mathbf{c}_i\|_2^2 \tag{4.19}$$

$$= \frac{1}{n_i}\sum_{j=1}^{n_i}(\|\widehat{\mathbf{x}}_{ij} - \widehat{\mathbf{c}}_i\|_2^2 + \|\mathbf{V}^T\mathbf{x}_i\|_2^2) \tag{4.20}$$

$$= \widehat{Vol}_i + \frac{1}{n_i}\sum_{j=1}^{n_i}\|\mathbf{V}^T\mathbf{x}_{ij}\|_2^2 \tag{4.21}$$

Therefore, the term $\frac{1}{n_i}\sum_{i=1}^{n_i}\|V^T\mathbf{x}_i\|_2^2$ measures the amount of volume shrinkage of a data group after the ASP reduction. The larger value this term has, the smaller volume a data group has in the reduced space, and the ASP reduction is more effective in the sense that data in the reduced space show a more compact clustering structure. We know that

$$\widehat{\mathbf{x}}^{\perp} = \mathbf{V}^T\mathbf{x} = \mathbf{V}^T(\mathbf{u}+\mathbf{v}) = \mathbf{V}^T\mathbf{v} \tag{4.22}$$

Then,

$$\frac{1}{n_i}\sum_{i=1}^{n_i}\|\mathbf{V}^T\mathbf{x}_i\|_2^2 = \frac{1}{n_i}\sum_{i=1}^{n_i}\|\mathbf{V}^T\mathbf{v}_i\|_2^2 \tag{4.23}$$

This means that for a fixed matrix $\mathbf{V}$, if data items $\mathbf{x}_i$ contain larger portion of structure-irrelevant information $\mathbf{v}_i$, the ASP method is more effective. The result is reasonable, since in such a case, more noise is removed in the reduced space.

On the other hand, when data $\mathbf{x}_i$ are fixed, matrix $\mathbf{V}$ will impact the performance of the ASP method. Since $\mathbf{V}$ is constructed from the representative matrix $\mathbf{C}$, where each column is a rank-1 approximation to a data group, the way we approximate a data group influences the performance. Given a data group $B_i$ with $n_i$ data items in $f$-dimensional space, our choice of using centroid to approx-

imate $B_i$ is based on the following fact the centroid $\mathbf{c}_i \in \mathbb{R}^f$ is the optimal rank-1 approximation to $B_i$ [JPR01] in the sense

$$\mathbf{c}_i = \sum_{j=1}^{n_i} \|\mathbf{x}_{i,j} - \mathbf{c}_i\|_2^2 = \min_{\mathbf{y}\in\mathbb{R}^f} \sum_{j=1}^{n_i} \|\mathbf{x}_{i,j} - \mathbf{y}\|_2^2 \tag{4.24}$$

## 4.5 Finding Orthonormal Basis for Range

To find the orthonormal basis $\mathbf{U}$ of $range(\mathbf{C})$, where $rank(\mathbf{C}) = r \leq b \ll f$, two well-studied rank-revealing matrix factorization techniques can be adopted.

1. *Singular Value Decomposition (SVD)*: We compute the *reduced* SVD decomposition of $\mathbf{C} \in \mathbb{R}^{f\times d}$ as: $\mathbf{C} = \mathbf{U}_r \Sigma \mathbf{V}_r^T$, where $\mathbf{U}_r \in \mathbb{R}^{f\times r}$ is an orthonormal set of basis vectors that span $range(\mathbf{C})$. Thus $\mathbf{U}_r$ is our projection. The SVD basis has a useful property that it automatically orders the dimensions according to their importance. If we desire to project data to a lower than $r$-dimensional subspace, i.e., a $t$-dimensional subspace, $t < r$, we can construct the transformation matrix by using the first $t$ eigenvectors of $\mathbf{U}_r$.

2. *QR factorization*: The *reduced* QR decomposition of $\mathbf{C} \in \mathbb{R}^{f\times d}$ is computed as: $\mathbf{C} = \mathbf{QR} = \begin{bmatrix}\mathbf{Q}_r \mathbf{Q}_{f-r}\end{bmatrix} \begin{bmatrix}\mathbf{R}_r \\ 0\end{bmatrix} = \mathbf{Q}_r \mathbf{R}_r$. Then $\mathbf{U} = \mathbf{Q}_r \in \mathbb{R}^{f\times r}$ is the desired orthonormal basis used for the ASP method. Compared to SVD, QR

decomposition has the advantages of being computationally more efficient and requiring less storage. The properties make QR decomposition more suitable for handling large data sets.

Note that the representative matrix $\mathbf{C} \in \mathbb{R}^{f \times d}$ is sparse and is a "tall and thin" matrix, since $r \leq d \ll f$. Therefore, both the reduced SVD and reduced QR are significantly faster than the the corresponding full decompositions. Besides, the representative matrix $\mathbf{C}$ is easy to construct. Suppose that the average number of data items within each data group is $m$ and there are $b$ data groups, constructing $\mathbf{C}$ takes $m \times b = n$ flops (flops is Floating item Operations Per Second). These features of the representative matrix $\mathbf{C}$ make the ASP method computationally very efficient. At last, note that ASP *automatically* determines the dimensionality of the reduced space as the rank of $\mathbf{C}$.

## 4.6 Relation to Other Methods

ASP shares similarities with another dimension-reduction-based semi-supervised clustering method, *SCREEN* [TXZW07]. Moreover, ASP has close relation with the classical unsupervised dimension reduction technique Principal Component Analysis (PCA), the classical supervised dimension reduction technique Linear Discriminant Component Analysis (LDA), and the supervised dimension reduction method centroidQR [JPR01][HJP03].

Let $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]_{f \times n}$ be the matrix representation of data to be clustered, $b$

be the number of data groups according to constraints, $\mathbf{c}_i$ be the centroid of the $i$th group, and $\mathbf{c}$ be the global centroid. We can define three kinds of scatter matrices, which are the "within group/class scatter matrix" $\mathbf{S}_w$, "between group/class scatter matrix" $\mathbf{S}_b$, and the "overall scatter matrix" $\mathbf{S}_t$

$$\mathbf{S}_w = \sum_{j=1}^{b}\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{c}_j)(\mathbf{x}_i - \mathbf{c}_j)^T \quad (4.25)$$

$$\mathbf{S}_b = \sum_{j=1}^{b}\sum_{i=1}^{n}(\mathbf{c}_j - \mathbf{c})(\mathbf{c}_j - \mathbf{c})^T \quad (4.26)$$

$$\mathbf{S}_t = \sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{c})(\mathbf{x}_i - \mathbf{c})^T \quad (4.27)$$

$$\mathbf{S}_w + \mathbf{S}_b = \mathbf{S}_t$$

Note that, the compactness of a data group/class is determined by the *trace* of $\mathbf{S}_w$, $trace(\mathbf{S}_w)$, i.e., the summation of diagonal elements of $\mathbf{S}_w$. The separation between groups/classes is determined by $trace(\mathbf{S}_b)$. High quality clusters should have small $trace(\mathbf{S}_w)$ and large $trace(\mathbf{S}_b)$. The goal of a dimension reduction method is thus to find a projection $\mathbf{P}$ that minimizes $trace(\mathbf{P}^T\mathbf{S}_w\mathbf{P})$ and maximizes $trace(\mathbf{P}^T\mathbf{S}_b\mathbf{P})$. The objective of ASP and other dimension reduction methods can be expressed in terms of the scatter matrices as

$$LDA : \max trace(\frac{\mathbf{P}^T\mathbf{S}_b\mathbf{P}}{\mathbf{P}^T\mathbf{S}_w\mathbf{P}})$$

$$CentroidQR : \max trace(\mathbf{P}^T\mathbf{S}_b\mathbf{P}),\ s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I}$$

$$
\begin{aligned}
PCA \quad &: \quad \max trace(\mathbf{P}^T(\mathbf{S}_w + S_b)\mathbf{P}),\ s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I} \\
SCREEN \quad &: \quad \max trace(\mathbf{P}^T\widetilde{S}_b\mathbf{P}),\ s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I} \\
&\qquad \text{where } \widetilde{\mathbf{S}}_b \text{ is a weighted version of } \mathbf{S}_b \\
ASP \quad &: \quad \max trace(\mathbf{P}^T(\widetilde{\mathbf{S}}_w + \mathbf{S}_b)\mathbf{P}),\ s.t.\ \mathbf{P}^T\mathbf{P} = \mathbf{I} \\
&\qquad \text{where } \widetilde{\mathbf{S}}_w \text{ is the optimal approximation of } \mathbf{S}_w
\end{aligned}
$$

Compare the two semi-supervised methods SCREEN and ASP, ASP is better at preserving the overall data structure because of the $\widetilde{\mathbf{S}}_w$ term in the objective function. This advantage is also shown by the empirical evaluation.

## 4.7 Performance Evaluation

### 4.7.1 Data Description

The semi-supervised clustering based on ASP projection method is evaluated using two public available data sets: the Reuters-21578 document corpus[1], and the 20-Newsgroups 18828 version[2] document corpus. Both document corpora are among the most widely used data sets for document clustering/classification purposes. The Reuters corpus contains 21,578 documents which are manually grouped into 135 clusters. In evaluations, only documents with a *single* label are included to ensure unambiguous results. The Newsgroups corpus contains 18,828 documents.

[1] `http://www.daviddlewis.com/resources/testcollections/reuters21578/`
[2] `http://www.ai.mit.edu/people/jrennie/20Newsgroups/`, 18828 version

Each document has no header except 'From' and 'Subject' lines. The Newsgroups data set is more challenging for clustering purpose because some of the newsgroups are very closely related to each other. Each document is preprocessed by tokenization, stop-words removal, and stemming. Each document is then converted into a feature vector based on the processed words, without frequency cutoff.

9 data sets, as summarized in Table 4.1, are generated from the two corpora as follows

1. To evaluate the clustering performance on data sets with various number of clusters, without the lose of generality, Reu-k data sets are generated from the Reuters corpus, with the number of topics $k$ ranging from 2 to 7. For data set Reu-k with $k$ topics, first $k$ topics are randomly selected, then 100 documents from each topic are randomly sampled and mixed (for topics with less than 100 documents, all the documents are selected).

2. To evaluate the performance on data sets that show various levels of clustering difficulties, 3 data sets are generated from the 20-Newsgroups corpus. Each data set contains 300 documents randomly sampled from 3 newsgroups. In the 20-Newsgroups corpus, some of the newsgroups are very closely related to each other (thus are more difficult to be clustered), while others are highly unrelated. **News-dif**ficult consists of 3 newsgroups on very similar topics, **News-med**iocre consists of 3 newsgroups on related topics, and **News-eas**y consists of 3 newsgroups on totally unrelated topics (thus are easy to cluster).

Table 4.1: Summarization of data sets ($n$: # of data items, $f$: # of features, $k$: # of clusters)

| data set | topics | $n$ | $f$ | $k$ |
|---|---|---|---|---|
| Reu-2 | earn, trade | 200 | 1,305 | 2 |
| Reu-3 | coffee, sugar, earn | 300 | 1,503 | 3 |
| Reu-4 | jobs, reserves, coffee, grain, | 249 | 1,459 | 4 |
| Reu-5 | gnp, copper, money-fx, alum, jobs | 316 | 1,722 | 5 |
| Reu-6 | acq, interest, reserves, grain, money-fx, earn | 500 | 1,855 | 6 |
| Reu-7 | cocoa, sugar, reserves, cpi, gold, earn, gnp | 544 | 2,097 | 7 |
| News-dif | comp.windows.x, comp.os., ms-windows.mis, comp.graphics | 300 | 3,570 | 3 |
| News-med | talk-politics.misc, talk.politics.guns, talk.politics.mideast | 300 | 4,457 | 3 |
| News-eas | alt.atheism, sci.space, rec.sport.baseball | 300 | 4,038 | 3 |

### 4.7.2 Competing Methods

To test whether the ASP method can effectively employ constraints to improve clustering accuracy, ASP is compared against a total of five competing methods. All the five methods have been successfully applied to clustering high-dimensional data, including text data. First, ASP is compared to two of the most widely-used unsupervised clustering methods for high-dimensional data

- Spherical k-means (SPKM) [DM01] adopts the standard k-means algorithm to cluster the normalized unit-length items by using the cosine similarity as the proximity function. SPKM is particularly suitable for handling text data.

- Normalized Cut (NC) [SM97a] has been recognized as one of the best spectral clustering method.

Moreover, ASP is compared to three semi-supervised clustering methods that well represent the three major branches of semi-supervised clustering techniques.

- MPCK [BBM04b, BBM04a] integrates metric learning, constraint-enforcement, and constraint-guided initialization into a probabilistic framework of Hidden Markov Random field. MPCK has been shown to outperform other metric learning methods, such as the one by Xing et al. [XNJR03].

- Constrained Normalized Cut (CNC) [JX06, YS04] represents the graph-based semi-supervised techniques and has shown success in handling high dimensional data, such as clustering text data and segmenting images.

- SCREEN [TXZW07], recently proposed, exploits pairwise constraints through semi-supervised feature projection and is designed particularly for handling high dimensional data.

### 4.7.3 Effectiveness in Handling Constraints

This experiment demonstrates the effectiveness of the ASP method in adopting constraints to improve clustering accuracy. The ASP method is compared against SPKM and NC using different amount of constraints, which ranges from 100 pairs to 800 pairs. Constraints are generated by randomly sampling data pairs according to ground truth. For a fixed amount of constraints, the final performance score is obtained by averaging the scores from 20 independent test runs.

Table 4.2 shows the normalized mutual information (NMI) values on Reuters and Newsgroups data sets. In the table, the first row (with the label "SPKM") and the second row (with the label"NC") show the results generated by the traditional SPKM and NC methods, respectively. The remaining rows show the results generated by the ASP method under different amount of constraints. It is clear from the table that SPKM generates better data partitions on the Reuters data sets, while NC is more suitable for clustering the Newsgroups data sets. For all data sets, ASP always outperforms SPKM and NC with large margins. Furthermore, when more number of pairwise constraints are available to ASP, better clustering accuracy is achieved.

Table 4.2: $NMI$ of SPKM, NC, and ASP ($t$: # of constraints)

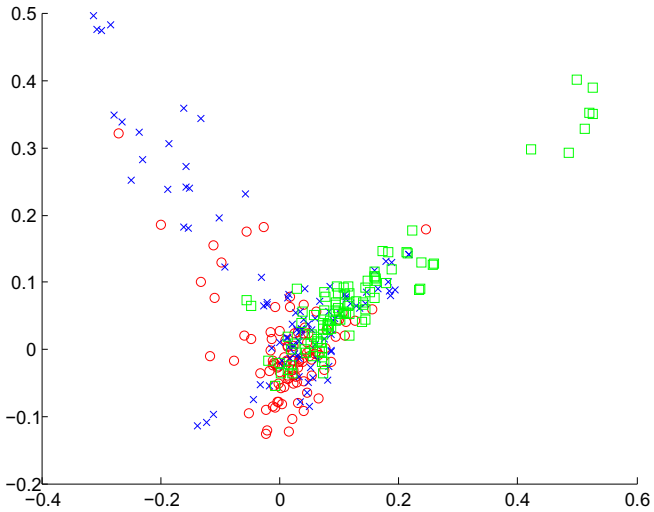
| Data Sets | $t$ | Reu-2 | Reu-3 | Reu-4 | Reu-5 | Reu-6 | Reu-7 | News-diff | News-med | News-eas |
|---|---|---|---|---|---|---|---|---|---|---|
| SPKM | | 0.8803 | 0.8544 | 0.7174 | 0.8432 | 0.6417 | 0.9047 | 0.1052 | 0.3425 | 0.6942 |
| NC | | 0.6687 | 0.8076 | 0.6249 | 0.7132 | 0.6664 | 0.8193 | 0.1018 | 0.5473 | 0.9251 |
| ASP | 100 | 1 | 0.9150 | 0.8843 | 0.8657 | 0.6931 | 0.9309 | 0.1979 | 0.5953 | 0.9191 |
| | 200 | 1 | 0.9363 | 0.8861 | 0.9140 | 0.7058 | 0.9553 | 0.3584 | 0.8227 | 0.9532 |
| | 300 | 1 | 0.9490 | 0.8949 | 0.9142 | 0.7139 | 0.9393 | 0.3572 | 0.8180 | 0.9830 |
| | 400 | 1 | 0.9363 | 0.9324 | 0.9232 | 0.7657 | 0.9322 | 0.6927 | 0.9421 | 0.9532 |
| | 500 | 1 | 0.9363 | 0.9477 | 0.9537 | 0.7679 | 0.9514 | 0.8517 | 0.9294 | 0.9830 |
| | 600 | 1 | 0.9660 | 0.9454 | 0.9539 | 0.7811 | 0.9415 | 0.8658 | 1.0000 | 0.9532 |
| | 700 | 1 | 0.9830 | 0.9572 | 0.9509 | 0.8398 | 0.9583 | 0.8531 | 0.9660 | 0.9660 |
| | 800 | 1 | 0.9860 | 0.9572 | 0.9586 | 0.8589 | 0.9661 | 0.9234 | 0.9830 | 0.9830 |

### 4.7.4 Noise Reduction and Visualization

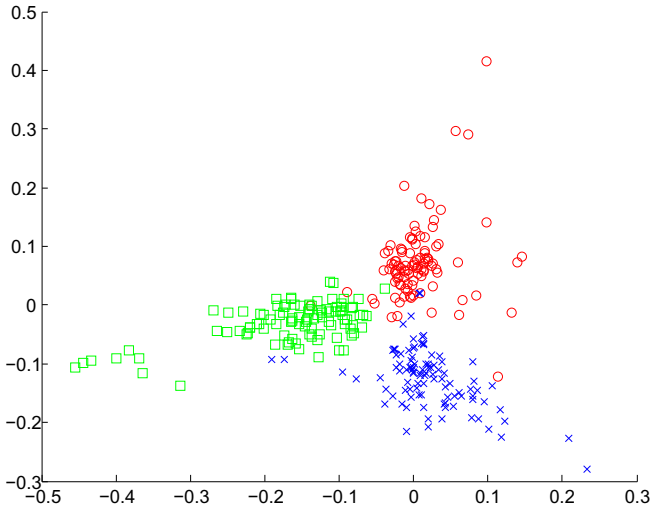
Given pairwise constraints, this experiment demonstrates that the ASP method can preserve the approximate-clustering-structure defined by constraints and remove structure-irrelevant noises in feature space. Thus ASP is useful for visualizing high dimensional data. Figure 4.1 shows the scatter plots of the News-dif data set before and after the ASP dimension reduction. The first 2 principal components are used to visualize the data clustering structure on a 2-d plan. Marker colors and shapes differentiate data items that belong to different clusters according to the ground truth. Figure 4.1(a) is the scatter plot before the ASP reduction. It is clear that data items from different clusters largely overlap. Besides, the within-cluster variance is larger. In other words, data items in the full-dimensional feature space do not show evident clustering structure. On the other hand, Figure 4.1(b) shows the scatter plot after ASP reduction with 800 pairs of constraints. Clusters in the reduced space are well separated, meanwhile, data items that belong to the same cluster get closer to each other as indicated by the smaller within-cluster variance. It is clear from Figure 4.1(b) that data clustering structure in the reduced space is more evident.

### 4.7.5 Clustering Accuracy

To reveal the effectiveness of ASP method as a semi-supervised clustering technique, this experiment conducts clustering accuracy comparisons with (1) MPCK,




(a) Before



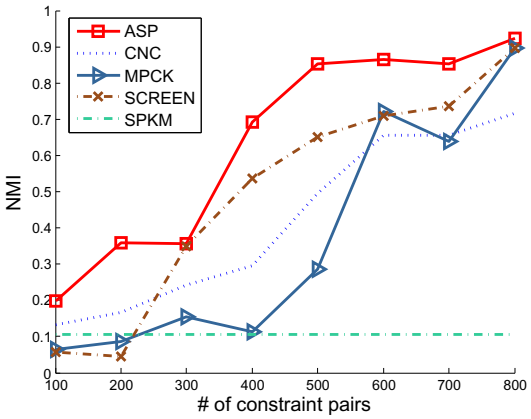

(b) After

Figure 4.1: Scatter plot of News-dif data set before and after the ASP method is applied (# constraints = 800).

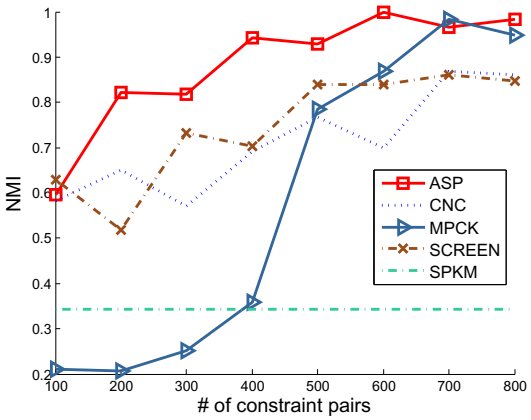
(2) CNC, and (3) SCREEN semi-supervised clustering methods. The three methods explore pairwise constraints through three different perspectives, and each represents the state of the art of the corresponding branch of semi-supervised clustering techniques. I used SPKM as the baseline. The results on the Newsgroups data sets are shown in Figure 4.2 and the results on the Reuters data sets are

shown in Figure 4.3 and Figure 4.4

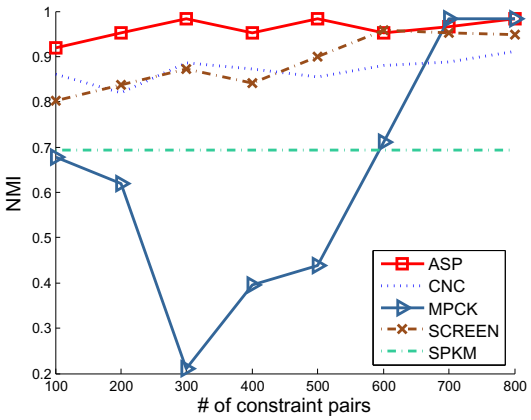
As Figures 4.2, 4.3 and 4.4 indicate, for all data sets with varying amount of constraints, overall, the ASP method yields better clustering accuracy than other semi-supervised methods. To further examine if this out-performance is statistically significant or not, the one-sided *Wilcoxon signed rank test* is conducted and the $p$-values between ASP and CNC, ASP and SCREEN, and ASP and MPCK pairs are calculated. The one-sided Wilcoxon signed rank test is a nonparametric paired test without assuming the underly distribution of the tested values. A sample pair is one pair of clustering accuracy values by two algorithms. Since the amount of constraint ranges from 100 to 800, there are 8 sampled pairs for each signed rank test. The results with $p$-value smaller than 0.05 are considered statistically significant. The test results are listed in Table 4.3. Except for the Reu-2 data set, the ASP method significantly outperforms other semi-supervised clustering methods. The relatively larger $p$-value on the Reu-2 data set is due to the fact that all the methods achieved accuracy equals 1 after the available constraints reached 400 pairs. Thus the difference between any two methods is not as significant as shown by other data sets.




(a) News-dif

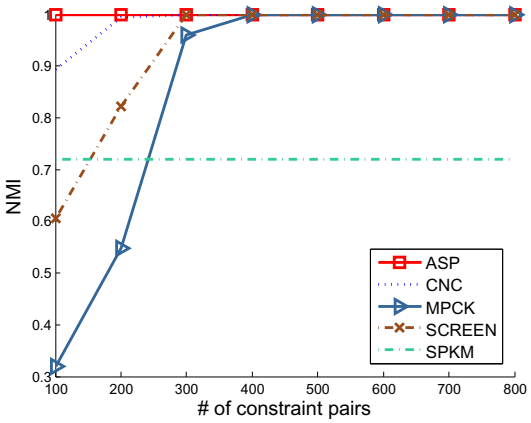



(b) News-med

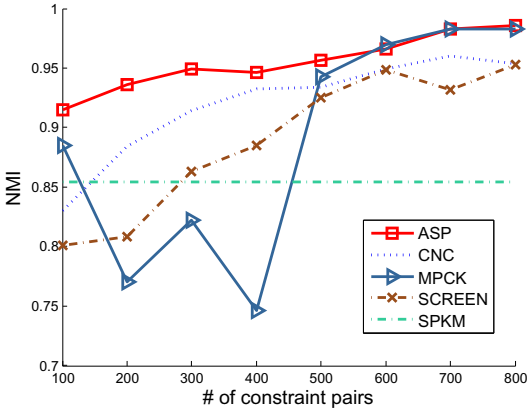



(c) *News-eas*

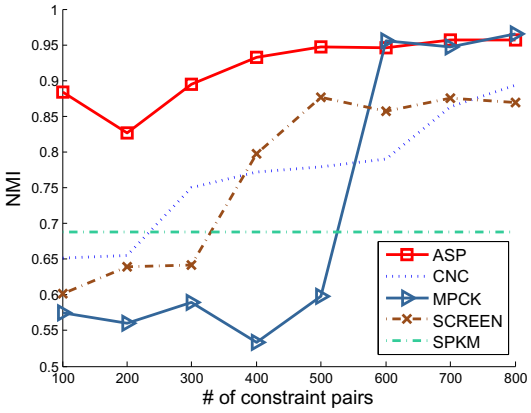
Figure 4.2: Accuracy comparison between ASP and other semi-supervised clustering methods on 20-Newsgroup corpus.




(a) Reu-2

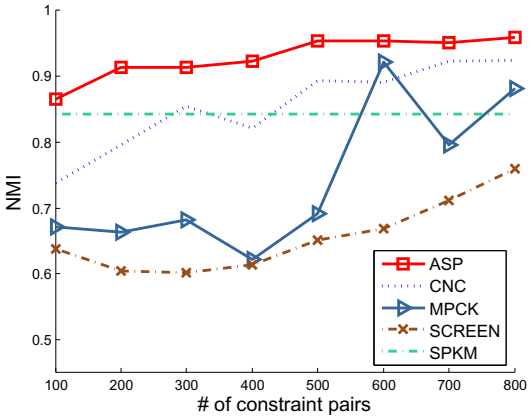



(b) Reu-3

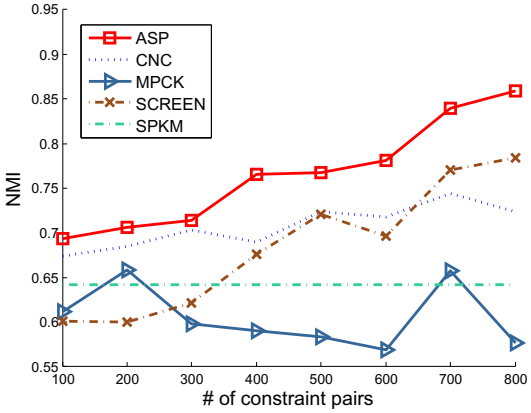



(c) Reu-4

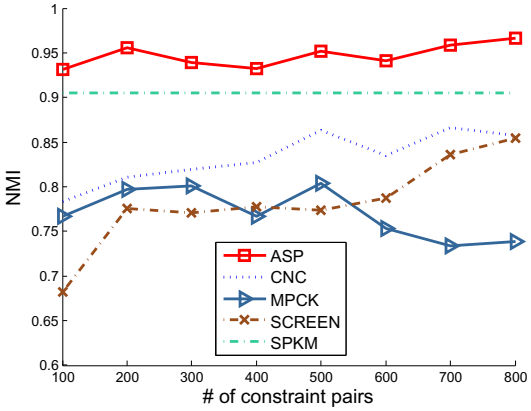
Figure 4.3: Accuracy comparison between ASP and other semi-supervised clustering methods on Reuters corpus.




(d) Reu-5




(e) Reu-6




(f) Reu-7

Figure 4.4: Accuracy comparison between ASP and other semi-supervised clustering methods on Reuters corpus (continue).

Table 4.3: $p$-value of the one-sided Wilcoxon singed rank test

| Data Sets | Reu-2 | Reu-3 | Reu-4 | Reu-5 | Reu-6 | Reu-7 | News-eas | New-med | News-dif |
|---|---|---|---|---|---|---|---|---|---|
| ASP vs. CNC | 0.25 | 0.0039 | 0.0039 | 0.0039 | 0.0039 | 0.0039 | 0.0039 | 0.0039 | 0.0039 |
| ASP vs. SCREEN | 0.25 | 0.0039 | 0.0039 | 0.0039 | 0.0039 | 0.0039 | 0.0078 | 0.0078 | 0.0039 |
| ASP vs. MPCK | 0.125 | 0.0273 | 0.0195 | 0.0039 | 0.0039 | 0.0039 | 0.0195 | 0.0078 | 0.0039 |

Besides the higher clustering accuracy, the ASP method is also more robust than other methods. Notice when the available amount of constraints is small (e.g., less than 400 pairs), the performance of CNC, MPCK and SCREEN can be inferior to that of the unsupervised method SPKM, while the ASP method still outperforms SPKM. The performance of ASP steadily increases when the amount of constraints increases, and always outperforms the unsupervised clustering method SPKM.

### 4.7.6 Computational Efficiency

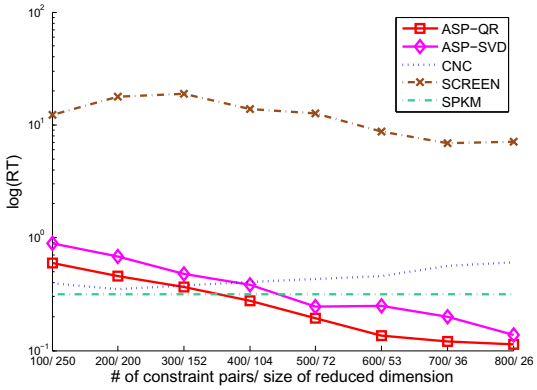
This experiment evaluates the computational performance of ASP, and compares it against the CNC and SCREEN methods. Again, SPKM is the baseline. The comparisons to the MPCK method are left out because MPCK is based on metric learning and its execution time is significantly longer than all other methods. Moreover, the computational performance is compared using different matrix factorization techniques in the ASP method. The ASP methods based on the SVD and QR decompositions are denoted as ASP-SVD and ASP-QR, respectively.

The computational performance is evaluated by measuring the running time (RT) of *1 run* for each algorithm on both the 20-Newsgroups data sets and the Reuters data sets. Experimental results on the 20-Newsgroups data sets are summarized in Figure 4.5 and results on the Reuters data sets are summarized in Figure 4.6 and Figure 4.7. Because the running time of different methods covers a large range of values, the logarithm scale for running time ($log(RT)$) is used for

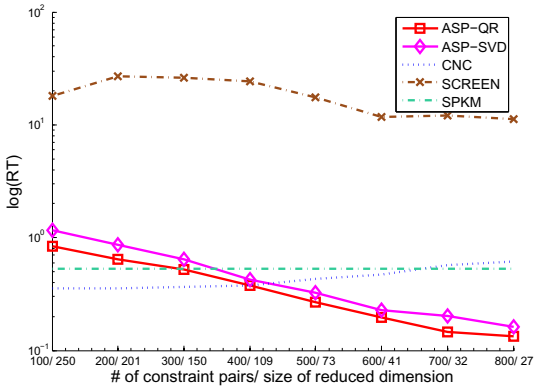
the $y$-axis to reduce the range and to get a better visualization.

As Figures 4.5, 4.6, and 4.7 indicate, for all the data sets, when the number of constraints increases, the running time of ASP decreases. This shows the nice scalability of ASP in exploiting constraints to aid the clustering process. Therefore, ASP can handle a large amount of constraints efficiently. Obviously, the largest running time of ASP is achieved when no constraint is available. After that, the more constraints are available, the shorter running time will be achieved, as well as the better clustering accuracy. Compared to the unsupervised SPKM method which clusters data in the full-dimensional space, when the amount of available constraints is small, due to the matrix factorization step of ASP, the running time of ASP is larger than SPKM. However, when more constraints are available, the representative matrix $\mathbf{C}$ becomes "thinner". So factorizing $\mathbf{C}$ is faster, and the running time of ASP becomes shorter than the running time of SPKM. Notice that, for all the data sets and various number of constraints, ASP-QR is always faster than ASP-SVD. This observation matches the fact that the QR orthogonal decomposition is more computationally efficient than the SVD decomposition.

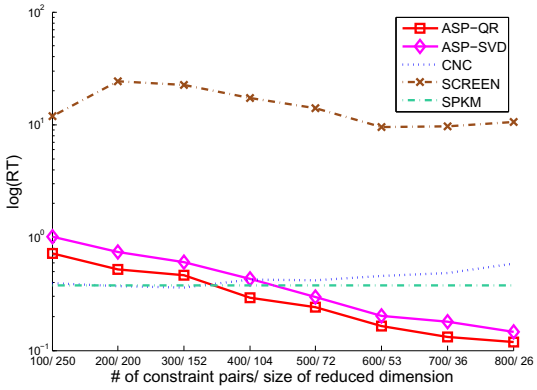
In contrast with ASP, the other dimension-reduction-based method SCREEN is computationally more expensive. Although SCREEN can also projects high-dimensional data onto a much-reduced space, where more efficient unsupervised data partitions can be performed, the scheme it employs to seek the desired data projection is computationally expensive (refer to [TXZW07] for details). The




(a) News-dif

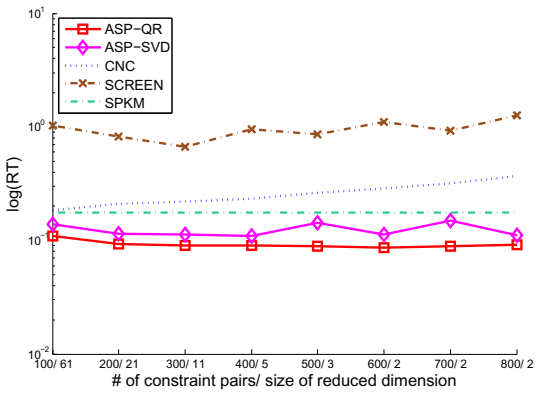



(b) News-med

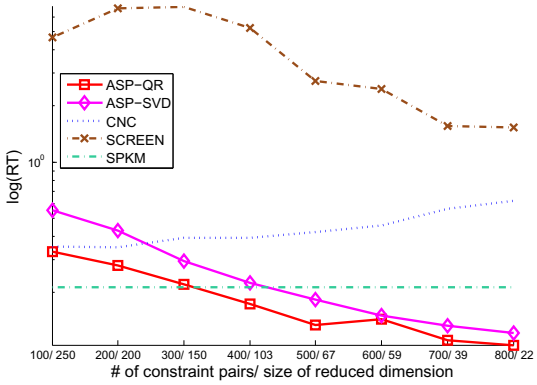



(c) News-eas

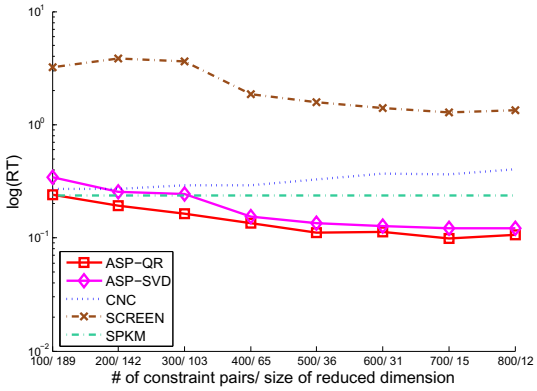
Figure 4.5: Running time comparison & subspace dimensionality on 20- Newsgroup corpus




(a) Reu-2

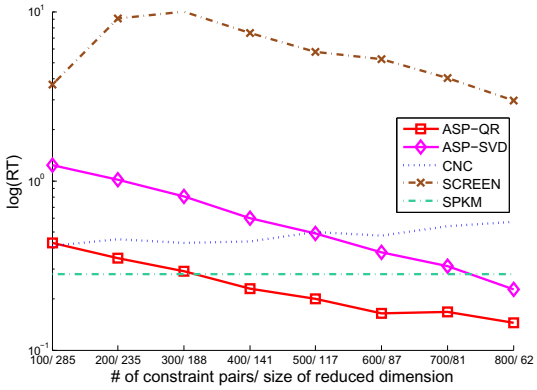



(b) Reu-3

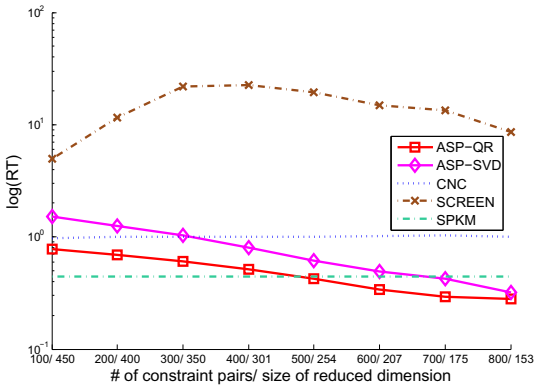



(c) Reu-4

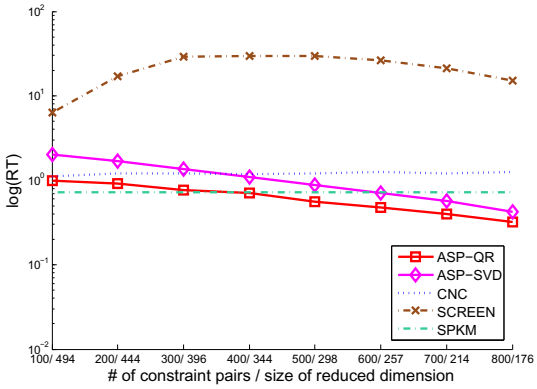
Figure 4.6: Running time comparison & subspace dimensionality on Reuters corpus.




(d) Reu-5




(e) Reu-6




(f) Reu-7

Figure 4.7: Running time comparison & subspace dimensionality on Reuters corpus (continue).

graph-based method CNC is a relatively efficient method. When the amount of constraints is small, CNC can be faster than ASP. However, when the amount of constraints increases, the running time of CNC increases too, and CNC becomes slower than ASP.

### 4.7.7 Dimensionality of the Reduced Space

Given a data set and pairwise constraints, the ASP method can automatically determine the dimensionality of the reduced space through rank-revealing matrix factorization techniques. And the dimensionality of the reduced space equals to the rank of the representative matrix. This feature of ASP sets a nice contrast to other dimension reduction techniques which often require large amount of repeated experiments to empirically determine the best dimensionality of the reduced space. Usually, the best dimensionality varies for different data set and in the semi-supervised clustering case, also varies for different amount of constraints. Therefore, potentially large amount of extra computations are required to determine the subspace dimensionality. ASP can save these computations. The dimensionality of the subspace found by ASP are shown in the $x$-axis of Figure 4.5, Figure 4.6 and Figure4.7.

### 4.7.8 Summary of Experiments

The finding can be summarized as follows: (1) As long as the constraints are provided, ASP always outperform the traditional SPKM and NC methods. (2) ASP can remove noise in the feature space to reveal evident clustering structure of the data set. (3) ASP significantly outperforms other semi-supervised clustering methods in clustering accuracy ($p$-value $< 0.05$). (4) ASP is computationally efficient. As the number of constraints increases, the running time of ASP decreases. Table 4.4 summarizes the comparison results between ASP and other semi-supervised methods. (5) ASP automatically determines the desired dimensionality of the reduced space.

Table 4.4: Summarization of experiments (Improvement by ASP in percentage)

| metric | Accuracy | | Running time | |
|---|---|---|---|---|
| data set | 20-Newsgroups | Reuters | 20-Newsgroups | Reuters |
| ASP vs. CNC | 32.90% | 9.43% | 15.16% | 51.86% |
| ASP vs. SCREEN | 53.36% | 19.29% | 97.84% | 94.08% |
| ASP vs. MPCK | 128.67% | 26.51% | faster | faster |

Chapter 5

# Semi-Supervised Clustering with Domain-Driven Noisy Constraints

In semi-supervised clustering, side information in the form of pairwise constraints can come from two sources. Constraints can be provided by human users and experts, or automatically identified based on domain knowledge. In most existing semi-supervised clustering approaches, the existence of well-defined noise-free constraints is often assumed, and semi-supervised approaches that use such constraints do not perform robustly when constraints contain noise. Noisy constraints, however, arise unavoidably in many real world applications. The noisy-constraints problem is particularly serious when constraints are automatically identified based on domain knowledge. This chapter studies the noisy-constraints problem in the domain of document clustering, although the technique is equally applicable to semi-supervised clustering in other domains.

## 5.1 Motivation

"Networked" documents proliferate with the development of the World Wide Web and Digital Libraries. In addition to text content attributes, networked documents are correlated by links (e.g., hyperlinks between Web pages, citations between scientific publications, and co-acted-by or co-directed-by relationships between movies etc.). These links are useful information for text analysis because they convey rich semantics that are usually independent of word statistics of documents [Hen05]. Among many successful techniques, PageRank [BP98] and HITS [Kle99] are two representative models which use the link information for document importance ranking.

Exploiting link information of networked documents to enhance text classification has been studied extensively in the research community [CDI98, CH00, GFKT03, OML00, TAK02]. It is found that, although both content attributes and links can independently form reasonable text classifiers, an algorithm that exploits both information sources has the potential to improve the classification [BEG06, Men04]. Similar conclusion has been drawn for text clustering by a growing number of works [AS06, BEG06, HZHDDS02, MS00, NAJ03, WK02, YL07]. However, the fundamental question still remains

*How to effectively couple the content and link information to get the most of both sources?*

Most of the previous studies couple content and link information for clustering

in one of the four following ways. Given documents represented by the bag-of-words model, the first way extends the term-based feature space with in-link and out-link features [MS00]. The second way linearly combines text similarity with link similarity [HZHDDS02, NAJ03]. The third way locally adjusts a document's cluster assignment based on its neighbors' cluster assignment in the link graph [AS06]. The fourth way weights terms according to link structure [BEG06]. Whereas these approaches provide valuable insights on employing link information, they either rely on heuristic combination of content and links, or assume the link graph to be dense or noise-free.

This chapter studies this problem for document clustering in a semi-supervised setting, and introduces a novel semi-supervised clustering approach for networked documents based on *COntent and STructure COnstrained (Costco) feature projection*. In particular, pairwise constraints are extracted from link structures and are further used to supervise the document clustering process. The link graphs of real-world data are usually sparse and noisy. Two link analysis methods are proposed to extract a small portion of links that are less noisy and are of higher probability to indicate topic correlations between documents.

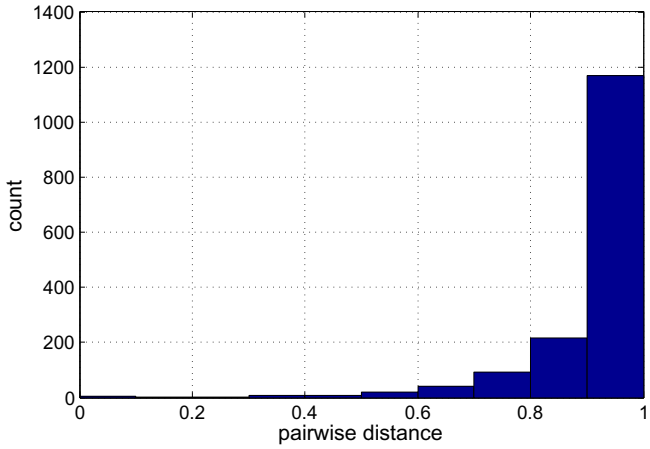
## 5.2 Problem Statement

There are two major issues with clustering networked documents. First, text data, usually represented by the bag-of-words model, have extremely high-dimensional

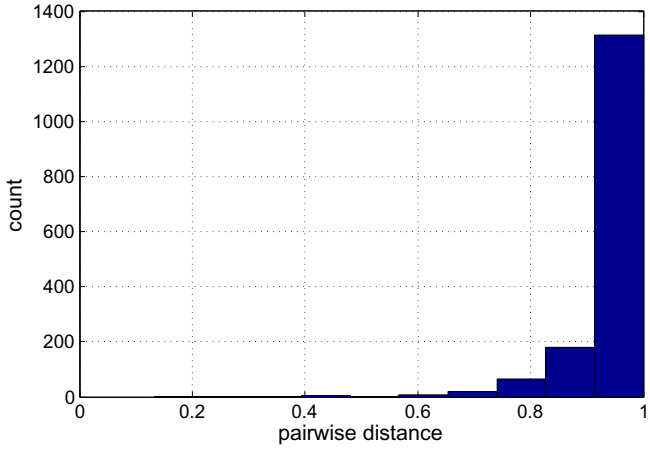
feature space (1000+). Thus, most document vetors tend to be equally far apart from each other, no matter they belong to the same topic or not. The effect is evident as shown in Figure 5.1. The distributions of pairwise distances are roughly the same for documents with the same topic and documents with different topics. An explanation of this phenomenon is that, the contribution of really discriminative words is marginal with respect to all the other words in the high-dimensional space. As introduced in Section 1.4, feature reduction is a better choice than feature selection in significantly reduce the dimensionality for text data.

Second, in the networked environment, semantically related documents tend to cite each other. If the link structure is noise-free and dense enough, then link-based clustering augmented by textual content [AS06, BEG06], will generally yield well separated clusters. However, the link structure is often noisy and sparse. For instance, many links in Web pages are for navigational purpose and therefore are not indicators of semantic relations [PT03]. Thus, it is crucial to find a text-based clustering solution that incorporates information from the available link structure as well.

The approach introduced in this chapter addresses the above two issues and bridges the disconnect between text and link structure from a feature projection perspective. An optimal projection direction is defined by satisfying constraints on both content and link structures. The low-dimensional data show more evident clustering structures and can be clustered with better quality.




(a) intra-cluster




(b) inter-cluster

Figure 5.1: Histograms of pairwise distances from two clusters (using 80 documents about topics *sci.med* and *comp.sys.ibm.pc.hardware* from 20-Newsgroups corpus).

## 5.3 Outline

The overall clustering framework is outlined in Figure 5.2. Given networked documents, two preprocessing steps are performed. On the one hand, link analysis is performed to extract *core pairs*, which are pairs of documents strongly correlated with each other according to the link structure. On the other hand, the traditional Vector Space Model (VSM) [BDJ99] is employed to convert documents into

high-dimensional vectors. Each dimension of a vector is a unique word after pre-processing (stopping, stemming, etc.). Core pairs and document vectors are then input into the feature projection module Costco. The generated low-dimensional data are partitioned by the traditional k-means clustering method into $k$ clusters, where $k$ is the desired number of clusters provided by users.

In the following sections, two link analysis methods that extract robust information from sparse and noisy link graphs are first introduced. Then, a novel feature projection method is introduced. The method takes use of the extracted link structure in searching for the optimal feature projection direction.

## 5.4 Link Analysis

The link graphs of real-world networked documents are usually sparse and noisy. This fact is illustrated with five real networked document data sets. Data sets *Cornell, Texas, Wisconsin*, and *Washington* are web pages collected from the four universities, while the data set *Cora* contains scientific articles that cite each other (refer to Section 5.7 for details). The small *average outdegrees* indicate the sparseness of the graphs. The statistic *informative links*% is defined as the percentage of links that the two connected documents by a link are about the same topic. Small percentages indicate that the link graphs are noisy. Therefore, instead of naively assuming a pair of connected documents being similar in topic, we need schemes to extract more robust link information from a link graph.

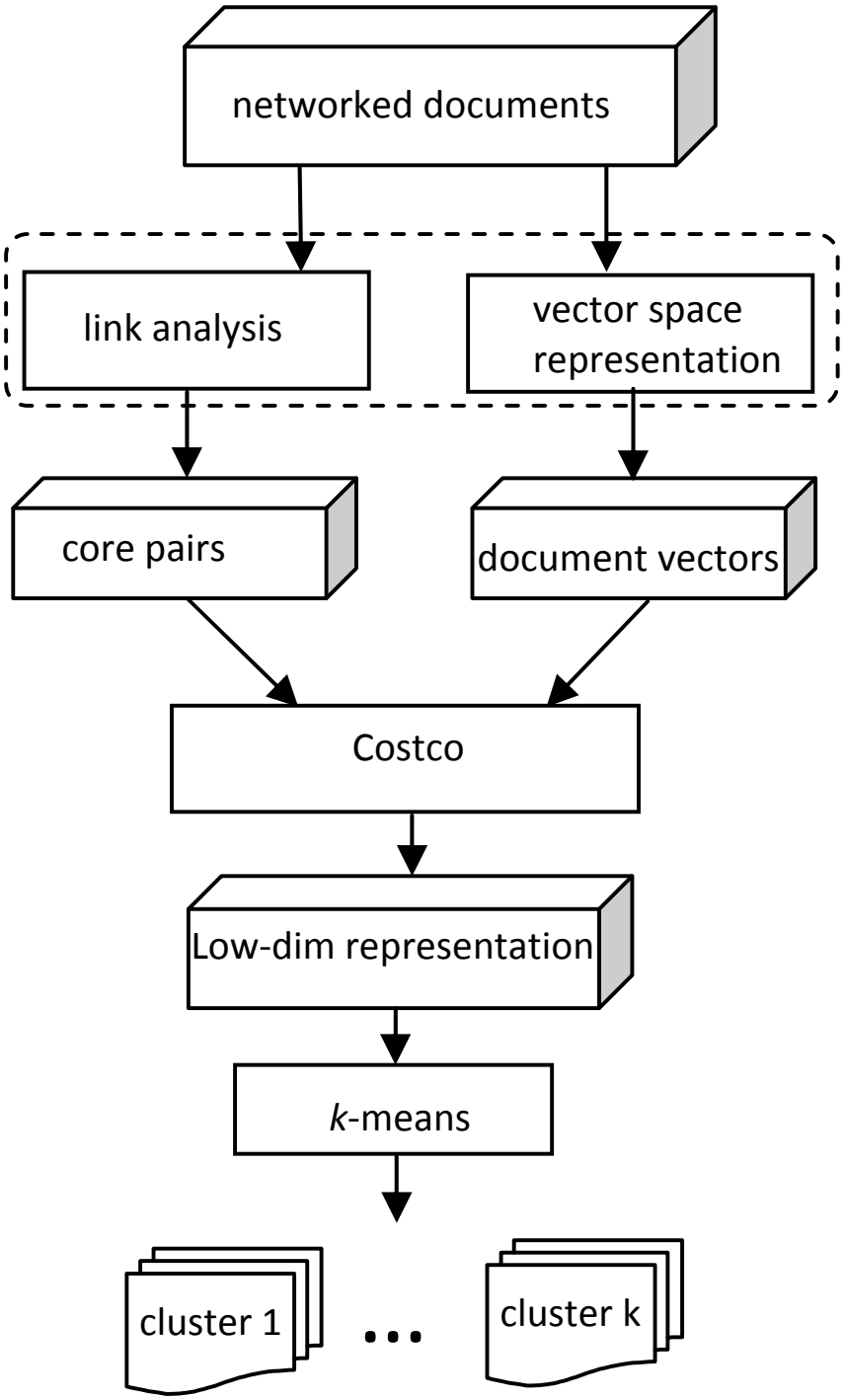



Figure 5.2: Framework of networked document clustering based on content and structure constrained feature projection (Costco)

Table 5.1: Link structure is sparse and noisy

| | Cornell | Texas | Wisconsin | Washington | Cora |
|---|---|---|---|---|---|
| avg. outdegree | 1.56 | 1.75 | 2.00 | 1.94 | 2.01 |
| informative links% | 13.16% | 11.58% | 20.38% | 24.89% | 81.38% |

### 5.4.1 Local Link Analysis

A link graph is modeled as *directed and unweighted*, denoted by $G(\mathbb{V}, \mathbb{E})$, where $\mathbb{V}$ is the set of the vertices/documents, and $\mathbb{E}$ is the set of edges/links between vertices. If document $\mathbf{d}_i$ links to/cites document $\mathbf{d}_j$, then there is an edge of unit weight starting from $\mathbf{d}_i$ and pointing to $\mathbf{d}_j$. Let matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$, where $n$ is the number of documents, be the corresponding *link matrix* defined as

$$l_{ij} = \begin{cases} 1 & \mathbf{d}_i \ \ cites \ \ \mathbf{d}_j \\ 0 & otherwise. \end{cases} \tag{5.1}$$

where $l_{ij}$ is the element on the $i$th row and $j$th column of $\mathbf{L}$. $\mathbf{L}$ embodies two types of document concurrences: *cociting* and *cocited*, as illustrated in Figure 5.3. A cociting relationship among a set of documents means that they all cite a same document. For example, both documents $\mathcal{A}$ and $\mathcal{C}$ cite document $\mathcal{D}$, so $\mathcal{A}$ and $\mathcal{C}$ have a cociting relation. A cocited relation refers to that several documents are cited together by another document. For example, document $\mathcal{B}$ and $\mathcal{D}$ are being cocited by documents $\mathcal{A}$. Both concurrences indicate the semantic correlations between documents.

In order to capture the concurrences, two adjacency matrices $\mathbf{X} \in \mathbb{R}^{n \times n}$ and

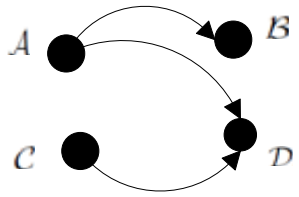



Figure 5.3: Cociting vs. Cocited

$\mathbf{Y} \in \mathbb{R}^{n \times n}$ are calculated

$$x_{ij} = \frac{|\mathbf{l}_{i*} \cap \mathbf{l}_{j*}|}{|\mathbf{l}_{i*} \cup \mathbf{l}_{j*}|}, \quad 0 \leq x_{ij} \leq 1 \tag{5.2}$$

$$y_{ij} = \frac{|\mathbf{l}_{*i} \cap \mathbf{l}_{*j}|}{|\mathbf{l}_{*i} \cup \mathbf{l}_{*j}|}, \quad 0 \leq y_{ij} \leq 1 \tag{5.3}$$

where $\mathbf{l}_{i*}$ and $\mathbf{l}_{*i}$ represent the $i$-th row vector and column vector of $\mathbf{L}$ respectively. $x_{ij}$ measures the Jaccard similarity of two documents $\mathbf{d}_i$ and $\mathbf{d}_j$ in terms of the cociting pattern, and $y_{ij}$ measures the similarity of the cocited pattern. Combining the two concurrences patterns, we have the overall structure-based similarity matrix

$$\mathbf{Z} = \alpha\mathbf{X} + (1 - \alpha)\mathbf{Y} \tag{5.4}$$

where $\alpha \in [0, 1]$ is the parameter that controls the contribution of each individual link pattern to the overall structure-based similarity. Note that, some previous work only consider the cocited pattern by heuristic, but ignore the cociting pattern [HZHDDS02]. It can be found that by exploiting both patterns, more robust information can be extracted from the noisy link graph. Given $\mathbf{Z}$, the set $\mathbb{C}$ of core

pairs is then defined as

$$\mathbb{C} = \{(\mathbf{d}_i, \mathbf{d}_j) | \mathbf{Z}_{ij} > \theta\} \tag{5.5}$$

where $\theta$ is a threshold that controls the reliability of link-based similarities.

### 5.4.2 Global Link Analysis

The link analysis scheme introduced in the previous section is a "local" method in the sense that for any query vertex/document in the graph, only the links between the query vertex and its direct neighbors are considered. Local analysis can miss some informative document pairs. Figure 5.4 shows such an example. According to the local method, documents $\mathcal{A}$ and $\mathcal{B}$ can be considered to be strongly connected since they both cite document $\mathcal{C}$. Similarly, we can find the document pair $\mathcal{C}$ and $\mathcal{D}$ since they both cite $\mathcal{E}$. However, the relations among documents $\mathcal{A}$, $\mathcal{B}$, $\mathcal{D}$ and $\mathcal{E}$ are ignored. Naively applying the transitive closer rule to link the four documents together may have the side effect of error diffusion if one of the judgement made by the local method is incorrect.

To this end, a global scheme is proposed that robustly finds all the strongly related document pairs in the link graph. In particular, a Markov random walk is defined on the link graph. Different from the local method, the link graph is modeled as *undirected and weighted*, denoted as $\tilde{G} = (\tilde{\mathbb{V}}, \tilde{\mathbb{E}})$. That means, if there is a link between two documents $\mathbf{d}_i$ and $\mathbf{d}_j$, a relation is considered to exits between them, no matter who starts the link. The edge is further weighted by the

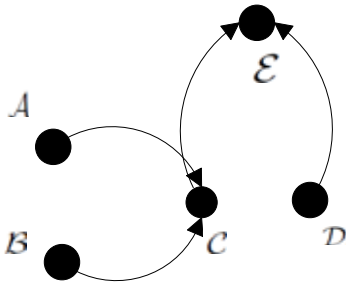



Figure 5.4: Local method misses informative pairs

pairwise similarity $\mathfrak{D}(\mathbf{d}_i, \mathbf{d}_j)$ of the two documents. Let matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$, where $w_{ij} = \mathfrak{D}(\mathbf{d}_i, \mathbf{d}_j)$, be the weight matrix. We then calculate the one-step transition probabilities $p_{ik}$, which are the probabilities of jumping from any state (vertex) $i$ to one of its adjacent state $k$, from these weights as

$$p_{ik} = \frac{w_{ik}}{\sum_j w_{ij}} \tag{5.6}$$

The one step transition probabilities can be organized as a matrix $\mathbf{P}$ whose $ik$-th entry is $p_{ik}$.

Due to the sparseness of a link graph, two documents that are strongly correlated in topics may not be linked together. For example, a scientific article can not cite all the related work, and several Web pages with similar topics may scatter in the Web without any link among them. See Figure 5.5 for an example. Suppose document $\mathcal{B}$ and document $\mathcal{D}$ are similar in topics (e.g., suppose $\mathcal{D}$ is in $\mathcal{B}$'s first $s$ nearest neighbors according to content similarity), but are not explicitly linked

together through any path in the graph. Similarly, suppose document $\mathcal{C}$, which is a singleton vertex, shares similar topics with document $\mathcal{A}$, but is completely unconnected. It is impossible, therefore, to identify all the core pairs from the link graph. To remedy this problem, for each vertex/document whose degree is below the average, artificial links are added between the vertex and its $s$ nearest neighbors where $s$ is a small number. For example, the dotted lines in Figure 5.5 denote the artificial links which establish correlations not originally conveyed by the sparse link graph.

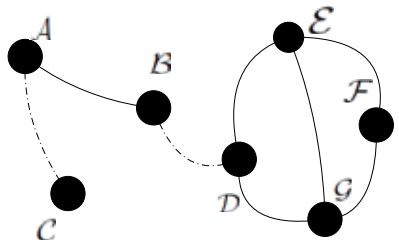



Figure 5.5: Sparse link graph misses informative pairs (real line: real links; doted line: artificial links)

For the augmented link graph, the transition matrix $\mathbf{P}$ has the property that $\mathbf{Pe} = \mathbf{e}$, i.e., $\mathbf{P}$ is stochastic, where $\mathbf{e}$ is the vector with all 1 elements. We can now naturally define the Markov random walk on the undirected graph $\tilde{G}$ associated with $\mathbf{P}$. The relation between two documents is evaluated by an important quantity in Markov chain theory, the *expected hitting time* $h(j|i)$, which is the expected number of steps for a random walk started at state $i$ to enter state $j$ for the first

time. Formally, $h(j|i)$ is defined as

$$\begin{cases} h(i|i) = 0 \\ h(j|i) = 1 + \sum_{k=1}^{n} p_{ik} h(j|k) \quad i \neq j. \end{cases} \tag{5.7}$$

The hitting time can be solved iteratively using the above recurrence relations or in closed form [AF95]. The choice of using expected hitting time to evaluate the correlation between two documents is justified by the desired property that the hitting time from state $i$ to state $j$ decreases when the number of paths from $i$ to $j$ increases and the lengths of the paths decrease. The core pairs can be naturally defined as

$$\mathbb{C} = \{(\mathbf{d}_i, \mathbf{d}_j) | (h(j|i) + h(i|j))/2 < \gamma\} \tag{5.8}$$

for a certain threshold $\gamma$.

## 5.5 Content & Structure Constrained Feature Projection (Costco)

This section introduces how to integrate content and link structure into a unified framework to find the optimal subspace embedding of high-dimensional data.

Let matrix $\mathbf{D} \in \mathbb{R}^{f \times n}$ be the document-term matrix where each column $\mathbf{d}_i$ is a document vector in the $f$-dimensional space. Let $\{(\mathbf{d}_{j1}, \mathbf{d}_{j2})\}|_{j=1}^{m}$ be the set of $m$ document pairs that have been identified as core pairs at the link analysis

step. Since these pairs of documents are strongly connected according to the link structure, there is a high probability that documents in a core pair are also semantically similar. We then prefer a projection direction, such that any two documents in a core pair will be *more similar* to each other after being projected along the direction, where the similarity between two documents is measured as

$$\mathfrak{D}(\mathbf{d}_{j1}, \mathbf{d}_{j2}) = 1 - \frac{\mathbf{d}_{j1}^T \mathbf{d}_{j2}}{\|\mathbf{d}_{j1}\| \|\mathbf{d}_{j2}\|} \tag{5.9}$$

To achieve this goal, we can minimize the variance between documents in a core pair. Let us define the scatter matrix $\mathbf{V}$ to encode the pooled variances for all the core pairs

$$\mathbf{V} = \frac{1}{m} \sum_{\{(\mathbf{d}_{j1}, \mathbf{d}_{j2})\} \in \mathbb{C}} (\mathbf{d}_{j1} - \mathbf{d}_{j2})(\mathbf{d}_{j1} - \mathbf{d}_{j2})^T \tag{5.10}$$

Then the desired projection is

$$\mathbf{S}^* = \arg\min_{\mathbf{S}} trace(\mathbf{S}^T \mathbf{V} \mathbf{S}) \tag{5.11}$$

where $\mathbf{S} \in \mathbb{R}^{f \times r}$ denotes the optimal transformation matrix, $r$ is the desired subspace dimensionality provided by users, and $trace(\cdot)$ is the trace of a square matrix, defined as the summation of the diagonal elements.

Directly minimizing Equation 5.11 leads to trivial solutions. For example, if the entire data set is projected to one point, then the covariance between core pair documents is minimized. To avoid trivial solution, we can put constrains on the

variance of the entire data set to prevent all the data points huddle together. The scatter matrix of the entire data set is defined as

$$\mathbf{U} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{d}_i - \mu)(\mathbf{d}_i - \mu)^T \tag{5.12}$$

where $\mu = \frac{1}{n}\sum_{i=1}^{n}\mathbf{d}_i$ is the global mean. Accordingly, the following objective is defined

$$\begin{aligned} \mathbf{S}^* &= \arg\max_{\mathbf{S}} trace(\frac{\mathbf{S}^T\mathbf{U}\mathbf{S}}{\mathbf{S}^T\mathbf{V}\mathbf{S}}) \\ &= \arg\max_{\mathbf{S}} trace((\mathbf{S}^T\mathbf{V}\mathbf{S})^{-1}(\mathbf{S}^T\mathbf{U}\mathbf{S})) \end{aligned} \tag{5.13}$$

which can be interpreted as

$$\arg\max_{\mathbf{S}} trace(\mathbf{S}^T\mathbf{U}\mathbf{S}) \quad and \quad \arg\min_{\mathbf{S}} trace(\mathbf{S}^T\mathbf{V}\mathbf{S}) \tag{5.14}$$

The objective function in Equation 5.13 defines a linear feature projection direction that both maximally preserves the variations of the entire data set and minimizes the pooled variances of core pairs. Simply put, after being projected along the optimal projection direction, the documents that are strongly connected (according to link structure) will be more similar to each other, while the rest documents are still well separated.

After the transformation matrix $\mathbf{S}$ is solved, the high-dimensional ($f$-

**Algorithm 3:** Costco (solving Equation 5.13). $\widehat{\mathbf{D}} = Costco(\mathbf{U}, \mathbf{V}, \mathbf{D}, k, r)$

**Input** : Scatter matrix $\mathbf{U}$;
Scatter matrix $\mathbf{V}$;
document-term matrix $\mathbf{D}$;
Desired # clusters $k$;
Desired # subspace dimension $r$.
**Output**: Low-dimensional data $\widehat{\mathbf{D}}$.

1 Do eigen analysis of $\mathbf{V} = \Phi\Delta\Phi^{\mathbf{T}}$;
2 Discard zero eigenvalues of $\mathbf{V}$ with their eigenvectors;
3 Form whitening transform $\mathbf{H} = \Phi\Delta^{-1/2}$;
4 Obtain new $\widetilde{\mathbf{U}} = \mathbf{H}^T\mathbf{U}\mathbf{H}$;
5 Do eigen analysis of $\widetilde{\mathbf{U}} = \Psi\Sigma\Psi^T$;
6 Pick the $r$ biggest eigenvalues $\lambda_1 \cdots \lambda_r$;
7 Pack their associated eigenvectors into the transformation matrix $\mathbf{S} = \mathbf{H} \times [\psi_1 \cdots \psi_r]$;
8 Project data via $\widehat{\mathbf{D}} = \mathbf{S}^T\mathbf{D}$;
9 **return** Low dimensional data $\widehat{\mathbf{D}}$.

dimensional) data can be optimally represented in the $r$-dimensional subspace as $\widehat{\mathbf{D}} = \mathbf{S}^T\mathbf{D}$, where $\mathbf{D} \in \mathbb{R}^{f \times n}$ is the original document-term matrix, $\widehat{\mathbf{D}} \in \mathbb{R}^{r \times n}$ is the subspace representation, $r \ll f$. The optimization problem 5.13 is a general eigenvalue problem. Detailed algorithm that solves Equation 5.13 and finds the transformation matrix $\mathbf{S}$ is shown in Algorithm 3.

## 5.6 Regularization

There are two factors that may prevent Costco from behaving robustly over a variety of data sets. First, if the covariance matrix $\mathbf{V}$ is singular and hence not invertible, the optimization problem in Equation 5.13 is ill-posed. Second, if a small number of core pairs are identified, the estimate of matrix $\mathbf{V}$ is highly variable due

to the randomness related to small sample size, which may lead to over-fitting.

Regularization is the technique to solve the ill-posed problem and to prevent overfitting. The regularized objective function of Costco is

$$\mathbf{S}^* = \arg\max_{\mathbf{S}} trace(\frac{\mathbf{S}^T\mathbf{U}\mathbf{S}}{\mathbf{S}^T(\mathbf{V}+\beta\mathbf{J})\mathbf{S}}) \tag{5.15}$$

where $\mathbf{J}$ is the regularizer, and the coefficient $\beta$ balances the model complexity and the empirical loss. A regularizer is what deemed to be more "physically plausible" to bias the sample-based estimate [Fri88]. The Tikhonov regularizer [Tik63], which is an identity matrix $\mathbf{J} = \mathbf{I}$, is usually adopted. For the document clustering problem, the following regularization term is proposed

$$\mathbf{J} = diag(cov(\mathbf{D})) \tag{5.16}$$

The hypothesis family of the diagonal regularizer preserves the differences in feature variances, thus is more realistic. Note that, if the data have been statistically normalized to unit variance, the diagonal regularizer is reduced to the Tikhonov regularizer since in such case $diag(cov(\mathbf{D})) = \mathbf{I}$. However, statistical normalization has the effect of making a data distribution "more Gaussian". When the true data distribution is quite different from Gaussian, normalization will impair data clustering accuracy.

In unsupervised learning, the regularization parameter $\beta$ is usually set to a fixed

**Algorithm 4:** Networked Document Clustering Based on Costco.

**Input** : A set of $n$ networked documents;
Desired # clusters $k$;
Desired # subspace dimensionality $r$.
**Output**: a set of clusters

```
begin link analysis
    Extract core pairs ℂ by local link analysis (Eq. 5.5)
    or global link analysis (Eq. 5.8)
begin content analysis
    Represent n documents using vector space model to get
    the document-term matrix D ∈ ℝ^{f×n};
Construct covariance matrix U (Eq. 5.12);
Construct covariance matrix V (Eq. 5.10);
Construct the regularizer J (Eq. 5.16)
and pick β (Eq. 5.17);
D̂ = Costco(U, V, D, k, r)
k-means(D̂, k);
return a set of clusters.
```

value by the algorithm designer without pellucid intuition [WZL07]. Instead, we can define the parameter as

$$\beta = \frac{\max(diag(\mathbf{V}))}{\max(diag(\mathbf{J}))} \tag{5.17}$$

where $\max(diag(\mathbf{A}))$ is the maximum value of the diagonal elements of matrix $\mathbf{A}$. The intuition is to scale the regularizer such that the sample-based estimate $\mathbf{V}$ and the regularizer $\beta\mathbf{J}$ are "comparable". That is, no one component will overwhelm the other. Empirical evaluation validates the effectiveness of this adaptive parameter setting scheme.

The overall clustering scheme is outlined in Algorithm 4.

## 5.7 Performance Evaluations

### 5.7.1 Data Description

The proposed networked document clustering framework has been evaluated on 6 UCI benchmark data sets[1], 3 data sets generated from the 20-Newsgroups document corpus, 3 data sets generated from the Reuters document corpus, the WebKB data sets[2] of hypertext, and the Cora data set[4] of scientific publications. All these data sets have been widely used for machine learning and text analysis tasks. Statistics of these data sets are listed in Table 5.2 to Table 5.5.

In particular, the 20-Newsgroups data sets are generated following the same steps as described in Section 4.7.1. To generate the Reuters data sets, the process is a little different from the process described in Section 4.7.1. For a given number of topics $b$, firstly, $b$ topics are randomly sampled, and then about 100 documents of each topic are randomly sampled and mixed together. Instead of fixing the $b$ topics, I randomly select $b$ topics for 5 times and generate 5 different data sets. In total, 15 Reu-k data sets have been generated. Table 5.4 shows the average statistics of 5 sets of independently generated data sets.

[1] `http://archive.ics.uci.edu/ml/`
[2] `http://www.cs.umd.edu/~sen/lbc-proj/LBC.html`

Table 5.2: UCI data sets

| data sets | # classes | # instances | # features |
|---|---|---|---|
| balance | 3 | 625 | 4 |
| vehicle | 4 | 846 | 18 |
| breast-cancer | 2 | 569 | 30 |
| sonar | 2 | 208 | 60 |
| ionoshpere | 2 | 351 | 34 |
| soybean | 4 | 47 | 35 |

Table 5.3: 20-Newsgroups data sets

| data sets | topics | # features |
|---|---|---|
| difficult | comp.windows.x, comp.os.ms-windows.mis, comp.graphics | 3,570 |
| mediocre | talk-politicis.misc, talk.politics.guns, talk.politics.mideast | 4,457 |
| easy | alt.atheism, sci.space, rec.sprot.baseball | 4,038 |

### 5.7.2 Baseline and Competing Methods

The spherical k-means (SPKM) [DM01] and the Normalized Cut (NC) [SM97a] [3] are chosen as baseline clustering methods. For competing dimension reduction techniques, the proposal is compared to two well-known unsupervised dimension reduction methods, the principal component analysis (PCA) [Pea01] which is a linear method and the locally linear embedding (LLE) [RS00][4] which is a non-linear

[3]original authors' implementation is used. `http://www.cis.upenn.edu/~jshi/software/`
[4]original authors' implementation is used `http://www.cs.toronto.edu/~roweis/lle/`

Table 5.4: Reuters data sets

| data sets | # classes | # instances | # features |
|---|---|---|---|
| reu4 | 4 | 400 | 2,537 |
| reu5 | 5 | 500 | 2,257 |
| reu6 | 6 | 600 | 2,626 |

Table 5.5: WebKB and Cora Data sets

| data sets | # classes | # instances | # features | # links |
|---|---|---|---|---|
| WebKB | 5 | 877 | 1,703 | 1,608 |
| Cora | 7 | 2,708 | 1,433 | 5,429 |

method. For competing techniques that couple content and link information, *Augmented* [MS00] and *L-Comb* [HZHDDS02, NAJ03] are implemented. *Augmented* augments the content-based vector space model with link features and applies k-means to the augmented document vectors. *L-Comb* linearly combines content similarity with link similarities and uses NC as the underlying clustering scheme. Other schemes of combining content and link information in clustering are available, such as local cluster membership adjustment [AS06] and link-based feature weighting [BEG06]. However, these techniques apply to dense or error-free link graphs so I do not compare to them. Besides, the method *Links* is a k-means clustering based on link similarity only. Table 5.6 summarizes all the algorithms that have been evaluated. The two proposals are bold faced.

### 5.7.3 Controlled Experiments

In this section, controlled experiments are performed to evaluate the effectiveness of various techniques in coupling content and links to improve clustering performance. In particular, given a data set, artificial links are generated and inserted between data points. In this way, the density of a like graph as the error rate of links are under control, and the me a method can be evaluated with various settings. Every

Table 5.6: Methods Summary

| method | description |
|---|---|
| FF(k-means) | spherical k-means (SPKM), baseline, content only, full feature space (FF) [DM01] |
| FF(NC) | Normalized Cut (NC), baseline, content only, full feature space (FF) [SM97a] |
| PCA | content only, reduced dimensionality [Pea01] |
| LLE | content only, reduced dimensionality [RS00] |
| Augmented | content & links, full feature space uses k-means for clustering [MS00] |
| L-Comb | content & links, full feature space uses Normalized Cut (NC) for clustering [HZHDDS02, NAJ03] |
| Links | links only |
| **Costco** | my proposal |
| **nr-Costco** | Costco without diagonal regularization |

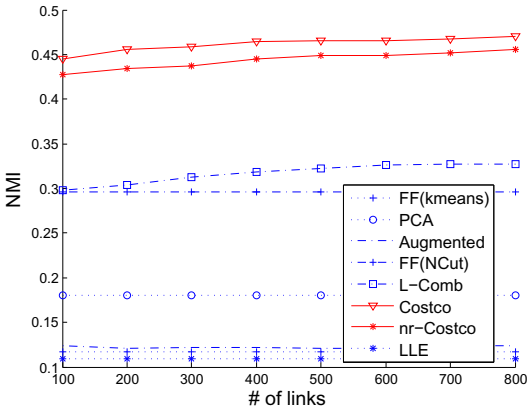
method that uses link information will take use of all the available links instead of pruning out some links with preprocessing steps. With controlled experiments, clustering schemes can be evaluated in a fair setting without being influenced by preprocessing.

To generate artificial links, the cluster membership relation of pairs of documents are randomly sampled from ground truth and $x$ pairs are chosen to add links in. Given an error rate $e$ of links, we can control the samples such that $\lceil x * e \rceil$ pairs of documents belong to different topics, which means these links are noise.

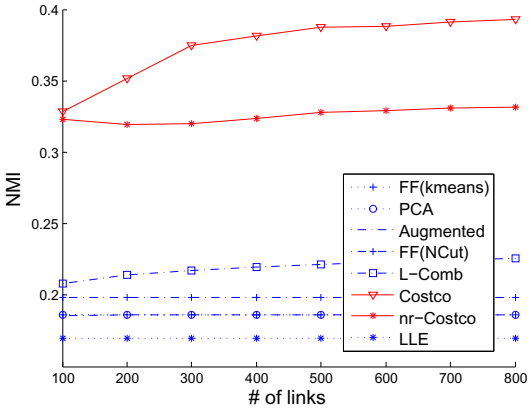
**Coupling Content and Links** In this experiment, the error rate of links is fixed to be zero, i.e., $e = 0$, and the graph density is varied by introducing $x$ =100 to 800 links between documents. This experiment measures the performance of a method in the noise-free setting with various levels of graph density.

Figures 5.6 5.7, 5.8 and 5.9 show the clustering performance measured by NMI for the UCI, 20-Newsgroups, and Reuters data sets respectively. Tables 5.7, 5.8 and 5.9 show the same result measured by RI and F score, with fixed 400 links. For all the data sets and different graph density levels, Costco consistently and significantly outperforms other competing methods. Notice that, L-Comb and Augmented improve clustering accuracy for some data sets e.g., *vehicle*, *balance*, *easy*, but do not consistently perform well for all the data sets.

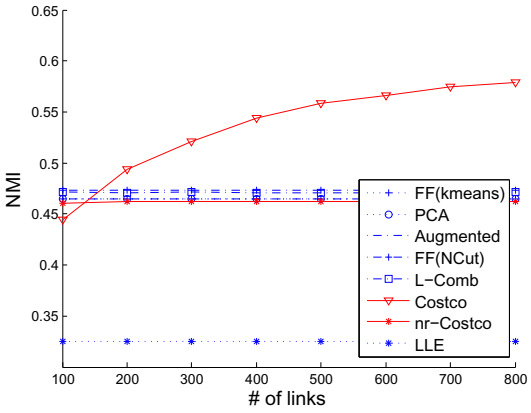
It is easy to observe that although the three evaluation metrics have different absolute values, they show very similar patterns for all the data sets and experiments.




(a) balance

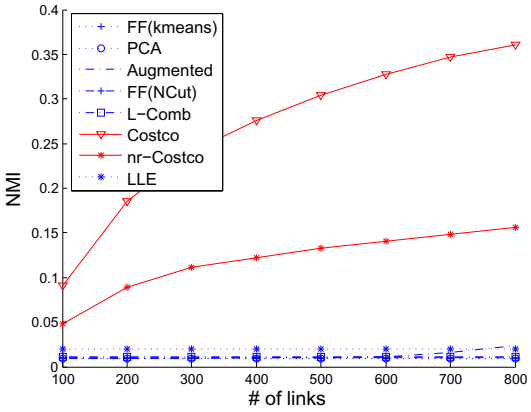



(b) vehicle

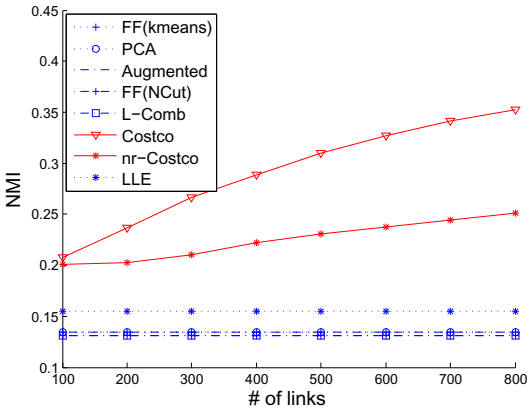



(c) breastcancer

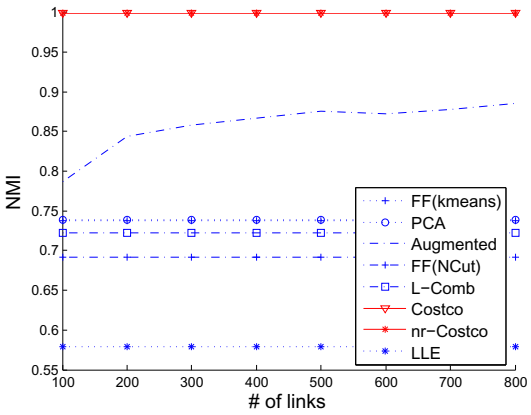
Figure 5.6: Clustering results on UCI data sets




(d) sonar

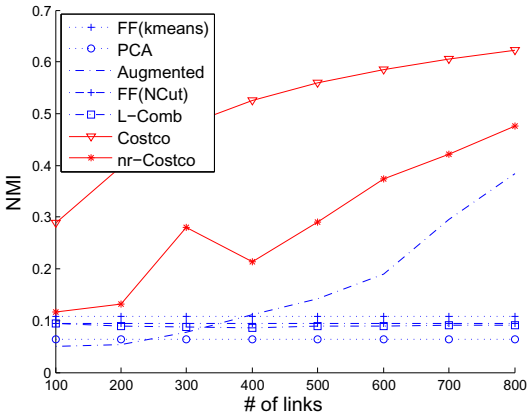



(e) ionoshpere

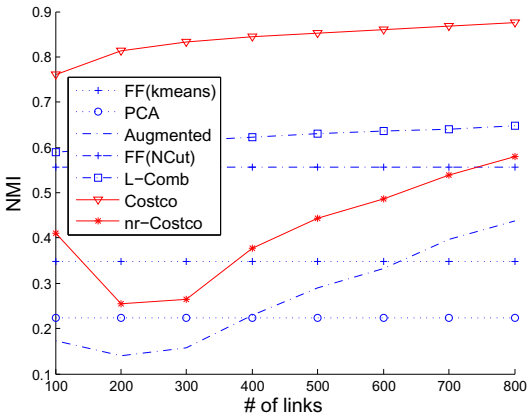



(f) soybean

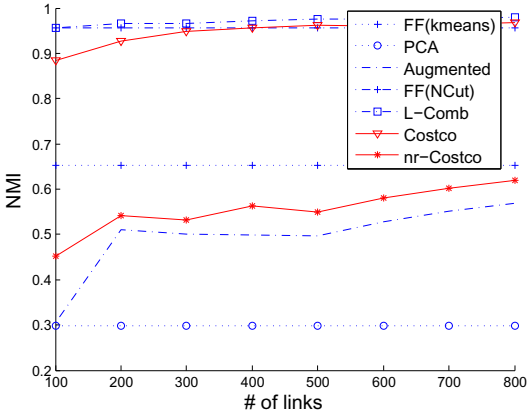
Figure 5.7: Clustering results on UCI data sets (continue)




(a) difficult

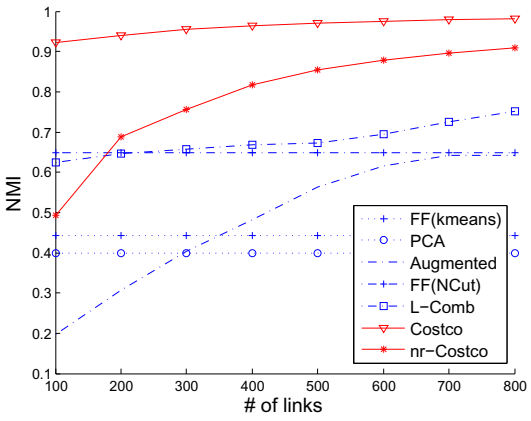



(b) mediocre

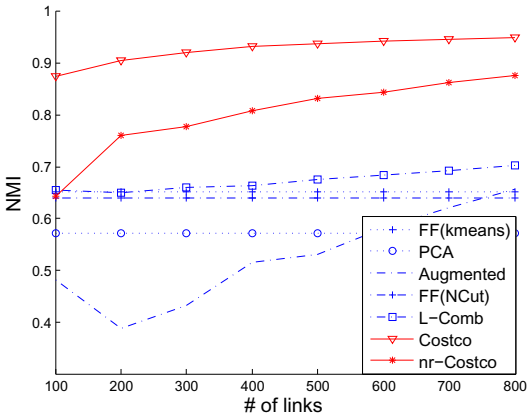



(c) easy

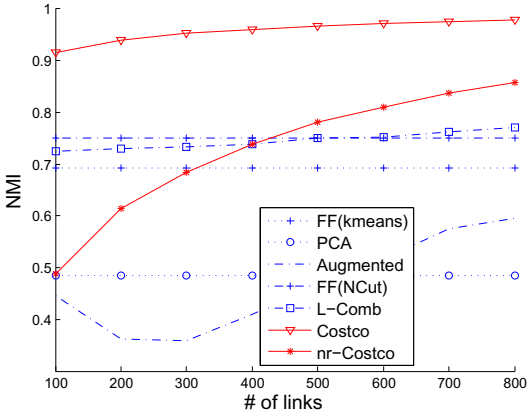
Figure 5.8: Clustering results on 20 Newsgroups data sets




(a) Reu4




(b) Reu5




(c) Reu6

Figure 5.9: Clustering results on Reuters data sets

Table 5.7: Performance on UCI data sets measured by RI and F (noise-free)(best results are bold-faced)

| Data sets | # of links | FF(kmeans) | PCA | LLE | Augmented | FF(NC) | L-Comb(NC) | **Costco** | **nr-Costco** |
|---|---|---|---|---|---|---|---|---|---|
| balance | 400 RI | 0.1806 | 0.6177 | 0.5730 | 0.5911 | 0.6706 | 0.6772 | **0.7151** | 0.7132 |
| vehicle | | 0.6462 | 0.6408 | 0.6507 | 0.6431 | 0.6709 | 0.6761 | **0.7404** | 0.7180 |
| breast-cancer | | 0.7504 | 0.7504 | 0.6356 | 0.7504 | 0.7554 | 0.7541 | **0.8008** | 0.7486 |
| sonar | | 0.5032 | 0.5032 | 0.5031 | 0.5041 | 0.5043 | 0.5046 | **0.6700** | 0.5749 |
| ionosphere | | 0.5889 | 0.5889 | 0.5933 | 0.5889 | 0.5841 | 0.5841 | **0.6509** | 0.6196 |
| soybean | | 0.8283 | 0.8291 | 0.7761 | 0.9065 | 0.8372 | 0.8372 | **1.0000** | **1.0000** |
| balance | 400 F | 0.4629 | 0.5010 | 0.4506 | 0.4658 | 0.5686 | 0.5771 | **0.6290** | 0.6270 |
| vehicle | | 0.3616 | 0.3650 | 0.3597 | 0.3635 | 0.3594 | 0.3730 | **0.5365** | 0.4785 |
| breast-cancer | | 0.7878 | 0.7878 | 0.6520 | 0.7878 | 0.7914 | 0.7905 | **0.8330** | 0.7866 |
| sonar | | 0.5028 | 0.5028 | 0.6042 | 0.5064 | 0.5041 | 0.5048 | **0.6828** | 0.5945 |
| ionosphere | | 0.6049 | 0.6049 | 0.6580 | 0.6049 | 0.5997 | 0.5997 | **0.7346** | 0.7188 |
| soybean | | 0.6761 | 0.6805 | 0.5485 | 0.8282 | 0.6716 | 0.6716 | **1.0000** | **1.0000** |

Table 5.8: Performance on 20-Newsgroup data sets measured by RI and F (noise-free) (best results are bold-faced)

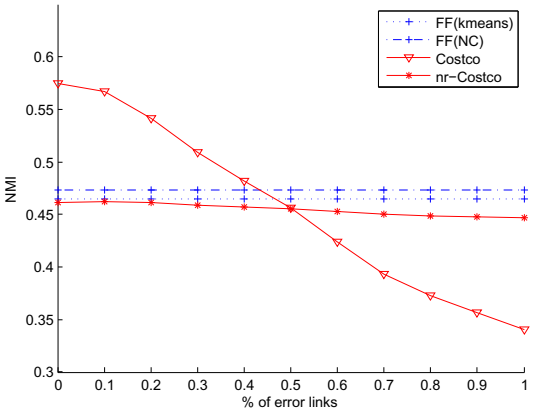
| Data sets | # links | FF(kmeans) | PCA | Augmented | (FF)NC | L-Comb(NC) | Costco | nr-Costco |
|---|---|---|---|---|---|---|---|---|
| difficult | | 0.5231 | 0.3910 | 0.4111 | 0.4493 | 0.4506 | **0.7868** | 0.5543 |
| mediocre | 400 | 0.5865 | 0.4579 | 0.4674 | 0.7105 | 0.7499 | **0.9375** | 0.6488 |
| easy | RI | 0.6858 | 0.2350 | 0.1610 | 0.9251 | **0.9431** | 0.9256 | 0.5565 |
| difficult | | 0.4424 | 0.4792 | 0.4786 | 0.4681 | 0.4660 | **0.7157** | 0.5444 |
| mediocre | 400 | 0.5299 | 0.4926 | 0.5088 | 0.6686 | 0.7072 | **0.9064** | 0.5978 |
| easy | F | 0.8375 | 0.4725 | 0.4725 | 0.9781 | **0.9833** | 0.9746 | 0.6370 |

Table 5.9: Performance on Reuters data sets measured by RI and F (noise-free) (best results are bold-faced)

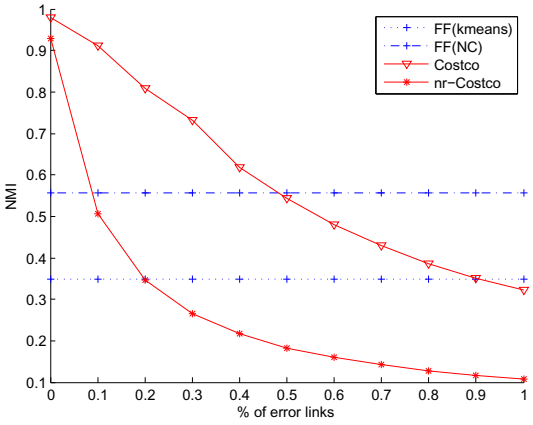
| Data sets | # links | FF(kmeans) | PCA | Augmented | (FF)NC | L-Comb(NC) | Costco | nr-Costco |
|---|---|---|---|---|---|---|---|---|
| Reu4 | | 0.6422 | 0.6694 | 0.6227 | 0.8141 | 0.8241 | **0.9891** | 0.8996 |
| Reu5 | 400 | 0.8172 | 0.7484 | 0.6626 | 0.8358 | 0.8405 | **0.9781** | 0.8973 |
| Reu6 | RI | 0.8563 | 0.6127 | 0.5433 | 0.9046 | 0.8791 | **0.9888** | 0.8809 |
| Reu4 | | 0.4932 | 0.5125 | 0.5297 | 0.6842 | 0.6977 | **0.9779** | 0.8323 |
| Reu5 | 400 | 0.6084 | 0.5285 | 0.4921 | 0.642 | 0.6493 | **0.9442** | 0.7761 |
| Reu6 | F | 0.6092 | 0.3966 | 0.3596 | 0.7240 | 0.6882 | **0.9657** | 0.6974 |

**Robustness to link errors** Follow a similar setting of the previous experiment, now the density of link graphs is fixed to have $x = 400$ links, and the error rate $e$ of links is varied from 0 to 1, with $e = 0$ means that there is no noise in links and $e = 1$ means that all the links are noise. Figure 5.10 shows the behavior of Costco for 3 representative data sets (results on other data sets show similar patterns and thus omitted). As long as most of the links are informative (i.e., the percentage of noisy links is below 50%), without any link-pruning preprocessing steps, regularized Costco always improves clustering accuracy. These results indicate the robustness of Costco to noisy link graphs.

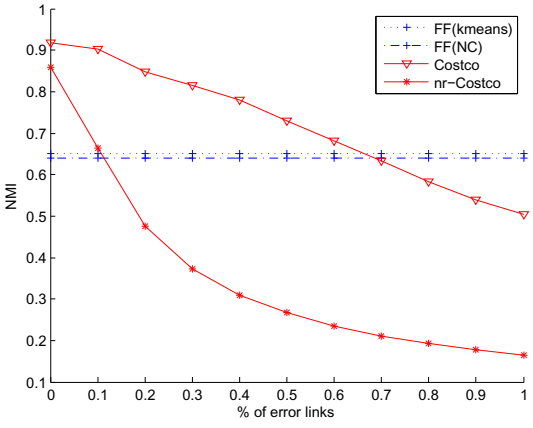
**Regularization** The effectiveness of the proposed diagonal regularizer can be easily observed from the reported experimental results. When no regularization is applied, doing dimension reduction based on noisy and sparse link structures may even degrade a clustering solution. For example, the mediocre data set (Figure5.8(b)) and the reu6 data set (Figure 5.9 (c)) show examples where the non-regularized Costco degrades clustering performance given small number of links. Figure 5.10 shows that without regularization, the use of noisy link structure in dimension reduction deteriorate clustering performance very fast. However, when the diagonal regularizer is adopted, the performance of clustering in the reduced space is much more robust. Note that, for all the data sets and experiments, the regularization parameter




(a) breast-cancer




(b) mediocre



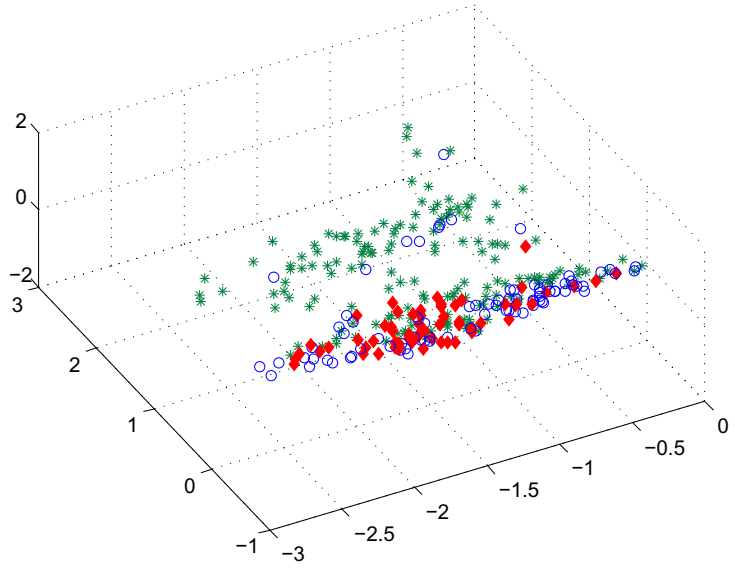

(c) reu5

Figure 5.10: Clustering results on Reuters data sets

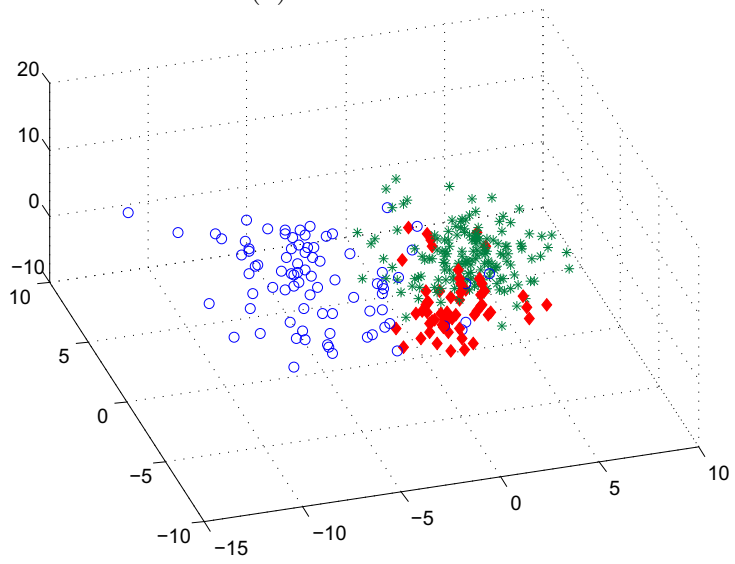
$\beta$ is set automatically and adaptively. The robust performance of Costco indicates the effectiveness of this parameter setting scheme.

**Dimension Reduction** In this experiment, for UCI data sets which have relatively low-dimensional features, the reduced dimensionality $r$ is fixed to be the half of the original dimensionality. For text data sets, the reduced dimensionality is set to 40 (this number does not change the relative performance comparison among competing methods). As reported results show, Costco always outperforms the other two unsupervised dimension reduction method, PCA and LLE. The performance gain is due to the use of link information. PCA and LLE, however, can not explore link information even when available. This experiment has shown that LLE does not perform well for text data sets, thus its results on text data is not reported here. This observation is due to the fact that LLE fails to handle sparse and weakly connected data such as text [SRS03]. Figure 5.11 shows the difference in data distributions after dimension reduction by PCA and Costco.

**Local vs. Global Link Analysis** In this experiment, instead of using all the available links, Costco adopts the local and global link analyses to extract robust core pairs of documents and does dimension reduction accordingly. With fixed 400 links and an error rate of 0.5, Figure 5.12 shows the clustering results. In most cases, both link analysis methods can prune noise in links and improve clustering performance. Global link analysis usually outperforms




(a) PCA reduction




(b) Costco

Figure 5.11: Dimension reduction by PCA and Costco on the Cora data set (Camera Position: [-49, -119, 241]).

local analysis as can be expected.

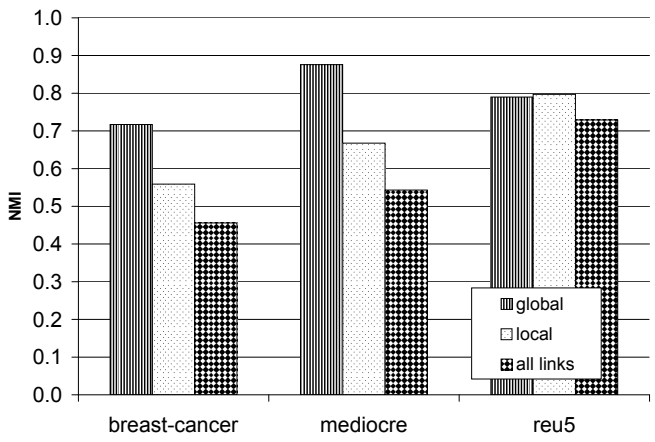



Figure 5.12: Link analysis: global vs. local methods

### 5.7.4 Unrestrained Experiments

Instead of artificially control the graph density and error rate, now all the methods are tested with real-world networked documents. Experimental results are shown in Table 5.10. Basically, similar patterns to the controlled experiments are observed. For example, in most cases, Costco outperforms competing clustering methods and dimension reduction methods. The regularization improves the robustness in clustering performance, and dimension reduction in general alleviates the curse-of-dimensionality problem related to text data and generates more accurate data partitions. Note that, because all the data sets have very sparse and noisy link structures (refer to Table 5.1), the clustering method *Links*, which entirely relies on link structures, has the worst performance. But when combining link structure with content information, all the three content and link coupling techniques improves clustering performance. This observation confirms the usability of link structure (can be sparse and noisy) in text analysis.

Table 5.10: Performance on Cora and WebKB data sets in NMI (best results are bold-faced)

| Data sets | *k*means | PCA | Costco | nr-Costco | Links | Augmented | L-Comb(NC) |
|---|---|---|---|---|---|---|---|
| Cornell | 0.2163 | 0.3058 | **0.3809** | 0.2054 | 0.1365 | 0.2105 | 0.3544 |
| Texas | 0.2276 | 0.3291 | 0.3755 | 0.2163 | 0.1643 | 0.3149 | **0.4121** |
| Wisconsin | 0.3977 | 0.4067 | **0.4846** | 0.2609 | 0.0977 | 0.3982 | 0.4592 |
| Washington | 0.3469 | 0.3352 | **0.3885** | 0.1599 | 0.1991 | 0.3221 | 0.3404 |
| Cora | 0.1361 | 0.1592 | **0.3712** | 0.1631 | 0.0336 | 0.1496 | 0.1817 |

Chapter 6

# Semi-Supervised Clustering of Non-linearly Separable Data

## 6.1 Motivation

This thesis has been focusing on semi-supervised clustering by linear transformations so far. Linear transformations perform well in general. However, when sever non-linearity is involved in data, linear transformations can be less effective.

One common and effective solution to the non-linearity problem is to use the popular kernel technique [STC04]. Kernel technique is based on the idea that non-linearly separable data can be separated linearly in some high-dimensional space. Data are first mapped to a high-dimensional feature space by non-linear transformations, then can be separated in the kernel-space with simple linear methods. Kernel technique has been applied to many machine learning and pattern recogni-

tion problems to improve performance. In terms of dimension reduction methods by linear transformations, most methods have their kernel space equivalents to deal with non-linearities in data. For example, kernel principal component analysis (PCA) [SSM97], kernel discriminant analysis(KDA) [MRW+99] , kernel metric multidimensional scaling (KMDS) [Web02], kernel locality preserving projections (KLPP) [HN03] and kernel relevant component analysis (KRCA) [TmCK05] are the kernel extensions to some representative dimension reduction approaches that are introduced in Chapter 2.

However, kernel methods also have some drawbacks compared to linear methods, which will be discussed in detail in Section 6.2. This chapter presents a semi-supervised clustering approach based on linear transformations, but the method is still able to cluster non-linearly separable data effectively. The method takes advantage of kernel technique but also preserves the simplicity of linear transformations.

## 6.2 Kernel Technique

A kernel method maps input data into a high-dimensional feature space using a nonlinear function, and if the mapping is chosen properly, complicate structures in the input space can be easily captured in the high-dimensional space [STC04]. Formally, let $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a positive definite kernel function satisfying for

all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$:

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \tag{6.1}$$

where $\phi$ is a nonlinear mapping function

$$\phi : \mathcal{X} \mapsto \mathcal{H} \tag{6.2}$$

that maps input space $\mathcal{X}$ into the $f_\phi$-dimensional feature space $\mathcal{H}$.

Kernel k-means [SSM98] is a representative kernel-based clustering technique. The traditional k-means has one major drawback that it cannot separate clusters that not non-linearly separable in the input space. Kernel k-means partitions the data points by linear separators in the new high-dimensional feature space.

Let $\pi_i$ denote the $i$th cluster of the total $k$ clusters, and a partitioning of data points be $\{\pi_i\}_{i=1}^k$. Kernel k-means generates a data partition by minimizing the following objective function

$$D(\{\pi_i\}_{i=1}^k) = \sum_{i=1}^{k} \sum_{\mathbf{x} \in \pi_i} \|\phi(\mathbf{x}) - \mathbf{m}_i\|^2 \tag{6.3}$$

where

$$\mathbf{m}_i = \frac{1}{|\pi_i|} \sum_{\mathbf{x} \in \pi_i} \phi(\mathbf{x}) \tag{6.4}$$

represents the centroid of cluster $\pi_i$ in the kernel space, and $|\pi_i|$ is the size of cluster $\pi_i$. As in traditional k-means, the kernel k-means assigns a data point to

the nearest cluster, where the point-to-cluster distance is measured as the point to cluster centroid distance in the kernel space.

To assign a data point to a cluster at each iteration, the Euclidean distance from the point $\phi(\mathbf{x})$ to centroid $\mathbf{m}_i$ needs to be calculated and is given by

$$\begin{aligned} \|\phi(\mathbf{x}) - \mathbf{m}_i\|^2 &= K(\mathbf{x},\mathbf{x}) - \frac{2}{|\pi_i|}\sum_{\mathbf{y}\in\pi_i} K(\mathbf{x},\mathbf{y}) \\ &\quad + \frac{1}{|\pi_i|^2}\sum_{\mathbf{y},\mathbf{y}'\in\pi_i} K(\mathbf{y},\mathbf{y}') \end{aligned}$$

The evaluation of the right-hand side of the above equation only involves the kernel function $K(\cdot,\cdot)$ and the input data points, thus can be solved in the kernel space.

It has been shown that kernel k-means is closely related to spectral clustering, which is a type of clustering technique that can handel non-linearities in data [DGK04]. Moreover, the objective function of normalized cut is identical to the objective function of weighted kernel k-means with the weights being selected in a particular way. Yet, kernel k-means is computationally more efficient than normalized cut for large data set [DGK04].

## 6.3 Motivation

The goal of this chapter is to cluster data items that are not linearly separable in the input space. For such data, the two types of constraints often lead to conflicting

data partitions, even if constraints by themselves are consistent. For example, this can lead to the over-constrained problem in semi-supervised k-means [WCRS01].

Kernel technique offers solutions to the non-linearity problem. However, despite the advantages as introduced in the previous section, kernel methods have some drawbacks

- How to properly choose the non-linear mapping is a big issue for kernel-based techniques. If the mapping is not chosen well, data in the kernel-space are not guaranteed to be linearly separable. For example, the clusters generated by kernel k-means can be of less quality than those generated by traditional k-means if the non-linear mapping is not chosen well.

- Kernel machines easily overfit. Any data that are not linearly separable in the input space is guaranteed to be separated linearly in some high-dimensional space. Thus, given pairwise constraints, it is always possible to find a data partition that satisfies all the constraints in certain high-dimensional space. However, kernel machine will overfit with limited constraints, since the non-linear mapping that satisfies a few pairs of constrains does not necessarily best reveal the structure in data.

- A kernel-based dimension reduction method is not easily generalized to handel new data. Because the mapping from the input space to the kernel space is non-linear, in order to map testing points, all the training points besides the transformation matrix need to be stored, and the inner product between

the testing points to all the training points need to be calculated and stored. The extra storage and computation cost limit the application of kernel-based dimension reduction methods to large data sets.

With the issues of kernel-based methods in mind, this chapter presents a semi-supervised dimension reduction technique based on Dual Subspace Projections (DSP). The approach can simultaneously preserve the structure of original high-dimensional data and the pairwise constraints specified by users. Thus, the method does not overfit. Furthermore, the method has a closed-form solution of an generalized eigenvalue problem, and therefore can be solved efficiently in the training phase. Moreover, the method uses kernel trick to handle nonlinearly separable data, but the learned projection is still linear. So handling testing data is very efficient. With the help of constraints, the method can also automatically identify the best hyperparameter in the kernel function.

## 6.4 Method Overview

Semi-supervised clustering by Dual Subspace Projections (DSP) is a two-step optimization solution. In particular, the must-link constraints are exploited in the first step by using a kernel subspace projection. At this step, a pair of must-link data points are mapped to one point in the kernel space, and are guaranteed to be assigned to a same cluster. In addition, the original distances are best preserved while satisfying constraints. Therefore, the structure of data after the first step

subspace projection is more informative for clustering analysis than the structure of the input space.

Since the structure of data after enforcing must-link constraints is fully encoded in the pairwise distances between data points in the kernel space, the method preserves only the pairwise distance information and moves on to the next step. In the second step, cannot-link constraints are further exploited to pull cannot-linked data points apart. This step also makes sure that the original distances are best preserved while satisfying constraints. An traditional k-means is then used to cluster data items after the two step subspace projections.

The two-step scheme solves the conflicting constraints problem. The method benefits from kernel technique in the first step and returns back to the original input space in the second step. The learned transformation is linear and can be easily generalized to handle new data.

## 6.5 Integrating Must-link Constraints and Data Structure

Given a pair of must-link constraint $(\mathbf{x}, \mathbf{x}')$, following the idea presented in [TPM09], the input space can be projected into the null space of the difference vector $(\mathbf{x} - \mathbf{x}')^T$, which is the direction orthogonal to the difference vector. Hence, $\mathbf{x}$ and $\mathbf{x}'$ will be mapped to a same single point, and the must-link constraint is

strictly satisfied. This method does not scale well with the increasing number of must-links. For data with $f$-dimensional features, if the number of must-link constraints exceeds $f-1$ all the data points will collapse to a single point. For this reason, I first map data to an enlarged feature space, i.e., the kernel space and then apply the same null space projection technique to explore must-link constraints. I call this method *kernel null space projection*. Figure 6.1 illustrates this idea using a one-dimensional data set.

Formally, define the $m \times f_\phi$ *must-link constraint matrix* $\mathbf{M}$ as follows

$$\mathbf{M} = \begin{bmatrix} (\phi(\mathbf{x}_1) - \phi(\mathbf{x'}_1))^T \\ \vdots \\ (\phi(\mathbf{x}_m) - \phi(\mathbf{x'}_m))^T \end{bmatrix} \tag{6.5}$$

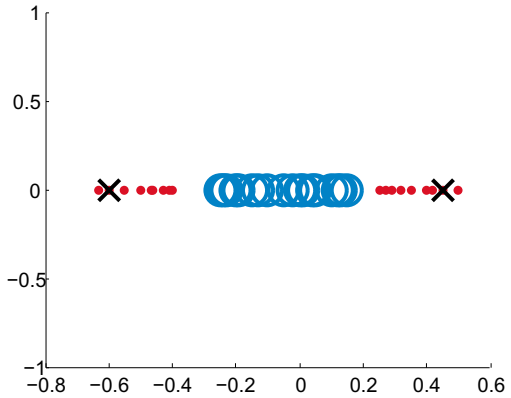
Then, the desired projection matrix is

$$\mathbf{P} = \mathbf{I}_{f_\phi} - \mathbf{U} \tag{6.6}$$

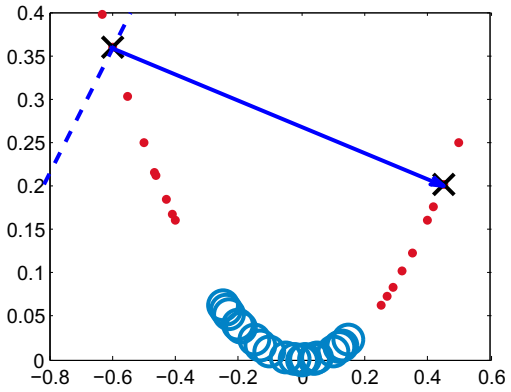
where

$$\mathbf{U} = \mathbf{M}^T(\mathbf{M}\mathbf{M}^T)^{\#}\mathbf{M}$$

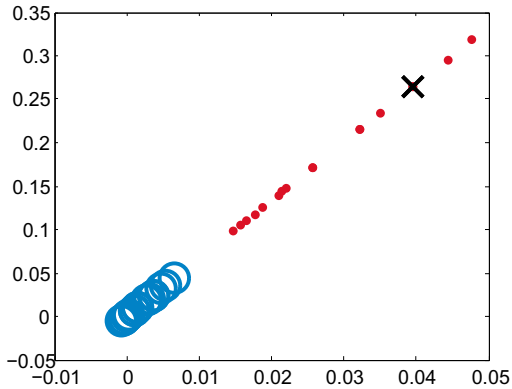
and # stands for the pseudo-inverse. $\mathbf{P}$ projects data in the feature space $\mathcal{H}$ to the null space of $\mathbf{M}$, and is the desired projection. One can prove that in the null space of $\mathbf{M}$, every pair of must-linked data points collapse to a single point, and thus the




(a)




(b)




(c)

Figure 6.1: Illustration of must-link constraints enforcement. (a) Input space. 36 one-dimensional data points originated from two clusters (18 points each, differentiated by markers) that are not linearly separable. Black crosses mark the must-link constraint pair $(\mathbf{m}_1, \mathbf{m}_2)$. (b) The input space is mapped to the 2-dimensional feature space via quadratic mapping $\phi(\mathbf{x}) = [\mathbf{x}\ \mathbf{x}^2]^T$. The blue arrow is the difference vector $(\phi(\mathbf{m}_2) - \phi(\mathbf{m}_1))^T$. The dotted line is the null space. (c) The feature space is projected to the null space of the difference vector. Constrained points collapsed to a single point and a clustering algorithm trivially groups them together.

must-link constraints are maximally satisfied (refer to Appendix for proof).

By simple algebra formulation, the projected kernel function is given by

$$\widehat{K}(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') - K(\phi(\mathbf{x}), \mathbf{M})^T \mathbf{W}^{\#} K(\phi(\mathbf{x}'), \mathbf{M}) \tag{6.7}$$

where $K(\phi(\mathbf{x}), \mathbf{M})$ denotes the $m$-dimensional vector

$$\begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) - K(\mathbf{x}, \mathbf{x}'_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_m) - K(\mathbf{x}, \mathbf{x}'_m) \end{bmatrix}$$

and

$$\mathbf{W}_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j) - K(\mathbf{x}_i, \mathbf{x}'_j) - K(\mathbf{x}'_i, \mathbf{x}_j) + K(\mathbf{x}'_i, \mathbf{x}'_j)$$

Since all the computations of $\widehat{K}(\mathbf{x}, \mathbf{x}')$ can be expressed in terms of $K(\mathbf{x}, \mathbf{x}')$, the subspace projection is performed implicitly in the kernel space.

Note that, the kernel null space projection $\mathbf{P}$ is the optimal projection in the sense that it preserves the variance along the orthogonal directions to the projection direction. Therefore, the original distances of data are best preserved while must-link constraints are satisfied.

## 6.6 Integrating Cannot-link Constraints and Data Structure

There are two kinds of data structures influence clustering accuracies. If data points from the same cluster are close to each other, the data set has better *intra-cluster structure*, which means a cluster is more compact. If data points from different cluster are far apart from each other, the data set has better *inter-cluster structure*, which means clusters are well separated.

The kernel null space projection introduced in the last section guarantees the enforcement of must-link constraints. Thus, the pairwise distances of the embedded data $d(\hat{\phi}(\mathbf{x}), \hat{\phi}(\mathbf{x}'))$ reveals the intra-cluster data structure better than the pairwise distances in the original space $d(\mathbf{x}, \mathbf{x}')$. However, the kernel null space projection can also mistakenly pull data points from different clusters close to each other, thus leading to clustering mistakes. Figure 6.2 illustrates this issue using the same data as in Figure 6.1 but with a different pair of must-link constraint. As a result, the pairwise distances of embedded data $d(\hat{\phi}(\mathbf{x}), \hat{\phi}(\mathbf{x}'))$ do not capture the inter-class structure well.

This problem can be solved by further exploiting cannot-link constraints based on the kernel null space projection result. The goal of adopting cannot-link constraints is to embed data in a subspace where data points from different classes are further pushed away from each other while the intra-class distance measure is

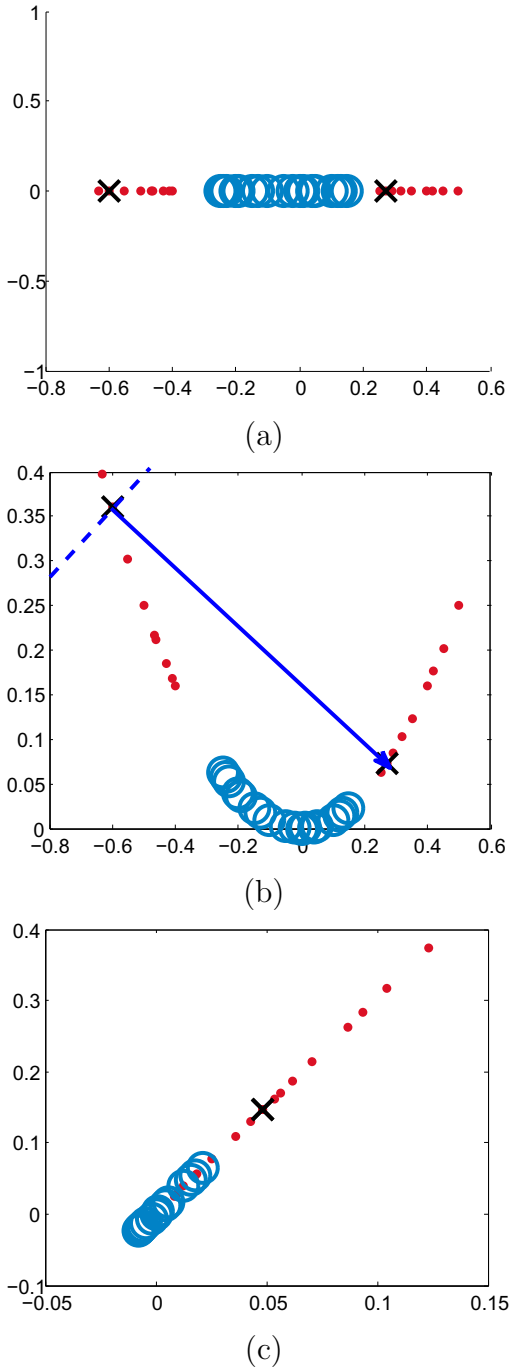



Figure 6.2: Illustration of a must-link enforcement error on unconstrained data points. Same set-up as Figure 6.1 with a different pair of must-link constraint. The null space projection result in (c) clearly demonstrates that although the constrained points are mapped to a single point, points from different clusters are mixed together too and leads to clustering mistakes.

still best preserved. Before presenting how to find such a subspace, I first make the following declaration and define a few concepts.

Without loss of generality, let us assume all the distances have been normalized to $[0, 1]$ in our discussion. Then the similarity between any two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is evaluated as $1 - d(\mathbf{x}_i, \mathbf{x}_j)$. Let $\mathbb{N}(\mathbf{x}_i)$ denotes the set of $k$-nearest neighbors of point $\mathbf{x}_i$ for a given $k$. Let $\mathbf{S}$ be the *adjacency matrix*, such that

$$\mathbf{S}_{ij} = \begin{cases} 1 - \hat{d}_\phi(\mathbf{x}_i, \mathbf{x}_j) & \mathbf{x}_i \in \mathbb{N}(\mathbf{x}_j) \vee \mathbf{x}_j \in \mathbb{N}(\mathbf{x}_i) \\ 0 & otherwise \end{cases} \tag{6.8}$$

where $\hat{d}_\phi(\mathbf{x}_i, \mathbf{x}_j)$ is the *kernel distance* defined as

$$\begin{aligned} \hat{d}_\phi(\mathbf{x}_i, \mathbf{x}_j) &= d(\hat{\phi}(\mathbf{x}_i), \hat{\phi}(\mathbf{x}_j)) \\ &= \sqrt{\widehat{K}(\mathbf{x}_i, \mathbf{x}_i) + \widehat{K}(\mathbf{x}_j, \mathbf{x}_j) - 2\widehat{K}(\mathbf{x}_i, \mathbf{x}_j)} \end{aligned} \tag{6.9}$$

and satisfies $\hat{d}_\phi(\mathbf{x}_i, \mathbf{x}_j) = 0$, if $(\mathbf{x}_i, \mathbf{x}_j) \in \Omega_M$. I adopt the kernel distances in the adjacency matrix because they fit the intra-class structure better.

Let $\mathbb{N}(\mathbf{x}_i)^\perp$ be the set of $k$ points that are farthest from $\mathbf{x}_i$ for a given $k$. Let $\mathbf{R}$ be a matrix which is called the *disjoint matrix*, such that

$$\mathbf{R}_{ij} = \begin{cases} 1 - d(\mathbf{x}_i, \mathbf{x}_j) & \mathbf{x}_i \in \mathbb{N}(\mathbf{x}_j)^\perp \vee \mathbf{x}_j \in \mathbb{N}(\mathbf{x}_i)^\perp \\ 0 & otherwise \end{cases} \tag{6.10}$$

Because the disjoint matrix mostly encodes the inter-class structure, the distance

measure of the original input space preserves the structure better.

Let $\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \dots & \mathbf{z}_r \end{bmatrix}$ be the matrix containing $r$ transformation vectors $\mathbf{z}_i|_{i=1}^{r}$ that embeds data points in the $f$-dimensional input space in the $r$-dimensional subspace by $\mathbf{y}_i = \mathbf{Z}^T\mathbf{x}_i$, $\mathbf{x}_i \in \mathbb{R}^f$, $\mathbf{y}_i \in \mathbb{R}^r$. In order to preserve both the intra and inter-class structures, I minimize the following objective function

$$\min \frac{\sum_{ij}(\mathbf{y}_i - \mathbf{y}_j)^2\mathbf{S}_{i,j}}{\sum_{ij}(\mathbf{y}_i - \mathbf{y}_j)^2\mathbf{R}_{i,j}} \tag{6.11}$$

The numerator incurs heavy penalties if nearby data points (i.e. $\mathbf{S}_{ij}$ is big) are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close then $\mathbf{y}_i$ and $\mathbf{y}_j$ are close as well. The denominator assigns big rewards if nearby data points from different classes (i.e. $\mathbf{R}_{i,j}$ is big) are mapped far away. Therefore, maximizing the denominator has the effect of pushing different classes farther away. Overall, minimizing Eq. (6.11) both preserves the structure of data and makes the structure more evident.

Similarly, the goal of pushing apart cannot-linked data points is achieved by maximizing the following objective function

$$\max \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\Omega_C} (\mathbf{y}_i - \mathbf{y}_j)^2(1 - d(\mathbf{x}_i, \mathbf{x}_j)) \tag{6.12}$$

If modify the disjoint matrix $\mathbf{R}$ to incorporate cannot-link constraints as

$$\widetilde{\mathbf{R}}_{ij} = \begin{cases} 1 - d(\mathbf{x}_i, \mathbf{x}_j) & \mathbf{x}_i \in \mathbb{N}(\mathbf{x}_j)^{\perp} \vee \mathbf{x}_j \in \mathbb{N}(\mathbf{x}_i)^{\perp} \\ & \vee \ (\mathbf{x}_i, \mathbf{x}_j) \in \Omega_c \\ 0 & otherwise \end{cases} \tag{6.13}$$

then the two objectives in Equation (6.11) and Equation (6.12) can be integrated into a single optimization problem as

$$\begin{aligned} \mathbf{z}^* &= \arg\min_{\mathbf{z}} \frac{\sum_{ij}(\mathbf{z}^T\mathbf{x}_i - \mathbf{z}^T\mathbf{x}_j)^2\mathbf{S}_{i,j}}{\sum_{ij}(\mathbf{z}^T\mathbf{x}_i - \mathbf{z}^T\mathbf{x}_j)^2\widetilde{\mathbf{R}}_{ij}} \\ &= \arg\min_{\mathbf{z}} \frac{\mathbf{z}^T\mathbf{X}\mathbf{L}_{\mathbf{S}}\mathbf{X}^T\mathbf{z}}{\mathbf{z}^T\mathbf{X}\mathbf{L}_{\widetilde{\mathbf{R}}}\mathbf{X}^T\mathbf{z}} \end{aligned} \tag{6.14}$$

where $\mathbf{L}_{\mathbf{S}} = \mathbf{D}^{\mathbf{S}} - \mathbf{S}$ and $\mathbf{L}_{\widetilde{\mathbf{R}}} = \mathbf{D}^{\widetilde{\mathbf{R}}} - \widetilde{\mathbf{R}}$ are the graph Laplacians [Chu97] related to the adjacency matrix $\mathbf{S}$ and the disjoint matrix $\widetilde{\mathbf{R}}$ respectively, and $\mathbf{D}^{\mathbf{S}}$ and $\mathbf{D}^{\widetilde{\mathbf{R}}}$ are diagonal matrices with $\mathbf{D}^{\mathbf{S}}_{ii} = \sum_j \mathbf{S}_{ij}$ and $\mathbf{D}^{\widetilde{\mathbf{R}}}_{ii} = \sum_j \widetilde{\mathbf{R}}_{ij}$. The $r$ optimal transformation vectors $\mathbf{z}_i^*|_{i=1}^r$ can be found by solving the general eigenvalue problem

$$\mathbf{X}\mathbf{L}_{\mathbf{S}}\mathbf{X}^T\mathbf{z} = \lambda\mathbf{X}\mathbf{L}_{\widetilde{\mathbf{R}}}\mathbf{X}^T\mathbf{z} \tag{6.15}$$

The $r$ eigenvectors related to the $r$ smallest eigenvalues are the solution.

Obviously, the performance of the above optimization problem strongly depends on the pairwise distances of data points, which are encoded in matrices $\mathbf{L}_{\mathbf{S}}$

and $\mathbf{L}_{\widetilde{\mathbf{R}}}$. By adopting the kernel distance $\hat{d}_{\phi}(\mathbf{x}_i, \mathbf{x}_j)$, and distances $d(\mathbf{x}, \mathbf{x}')$ of the original input space, the modification to the feature space in the kernel null space projection step is incorporated. Therefore, the final optimal projection direction is determined by both types of constraints as well as the intrinsic structure of data.

## 6.7 Performance Evaluation

### 6.7.1 Data Sets

Multiple real data sets from different domains are used to evaluate the performance of double subspace projections. Data sets are summarized in Table 6.1. The data sets used are very diverse in terms of size of data, size of feature spaces and number of clusters. In particular, 10 data sets are gathered from the UCI machine learning database [1] because of their popularity in the field of machine learning. Besides, the COIL-20 database [2] is used, which is widely used in 3D object recognition research. This database contains gray-scale images of 20 objects. Each object has 72 images taken at different orientations. Thus, the entire database contains 1,440 images. Each image is of size $128 \times 128 = 16,384$ pixels. I further perform bicubic interpolation to downsize every image to $16 \times 16$ pixels. This is a commonly used technique to achieve tradeoff between complexity and accuracy. Thus, each image is represented as a vector of dimension 256. Samples of the COIL-20 database are

[1] http://archive.ics.uci.edu/ml/

[2] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

Table 6.1: Data sets summary ($n$: # samples; $f$: # features; $k$: # clusters; $\delta$: kernel parameter )

| data set | $n$ | $f$ | $k$ | $\delta$ |
|---|---|---|---|---|
| wine | 178 | 13 | 3 | 0.6 |
| vehicle | 846 | 18 | 4 | 0.9 |
| iris | 150 | 4 | 3 | 0.3 |
| balance | 625 | 4 | 3 | 0.7 |
| ionosphere | 351 | 34 | 2 | 1 |
| glass | 214 | 9 | 6 | 0.3 |
| breast | 682 | 10 | 2 | 1 |
| Multiple Features | 2,000 | 649 | 10 | 0.2 |
| isolet | 7,797 | 617 | 26 | 7 |
| Pendigit | 10,992 | 16 | 10 | 46 |
| COIL-20 | 1,440 | 16,384 | 20 | 0.4 |

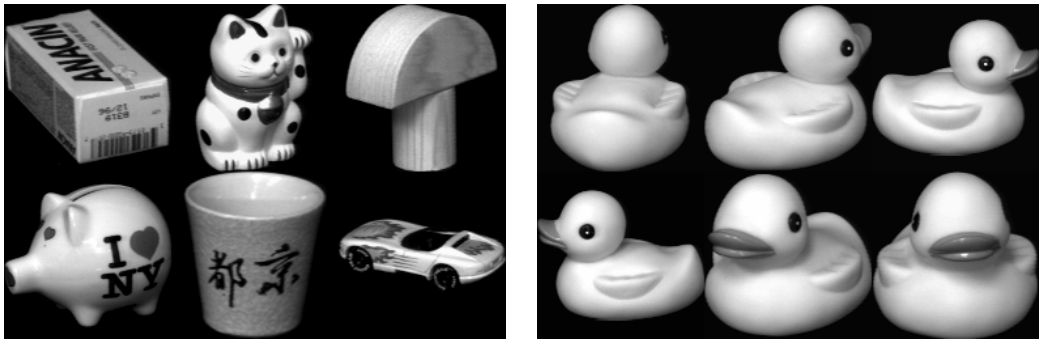


Figure 6.3: COIL-20 database. Left: 6 random samples, right: 6 orientations of one object

listed in Figure 6.3.

### 6.7.2 Competitive Techniques and Evaluation

The semi-supervised clustering method by dual subspace projections has been compared to four state-of-the-art and representative semi-supervised and unsupervised dimension reduction techniques. LPPSI [ALV08] is a recent semi-supervised dimension reduction method that has been successfully applied to solve face recog-

nition problem. The proposal is compared to the kernel version of LPPSI since it is reported to have better performance than the non-kernel version. LPP [HN03] is a dimension reduction technique that preserves the local structures of data, and has been widely adopted in visualization and text indexing. The proposal is also compared to SLPP, which is the supervised version of LPP that is able to explore constraints. PCA is the classical unsupervised dimension reduction technique that optimally preserves data variance. The dimension reduction performance achieved by each method is measured in a clustering setting and k-means is adopted as the underlying clustering model in all experiments. A better semi-supervised dimension reduction method should be able to take full advantage of side information and better reveal the intrinsic structure of the data, and thus leads to higher clustering accuracy. $F$-score is adopted to evaluate clustering accuracy. The clustering error rate is defined as $1 - F$-score. All the reported results are based on the average of 20 independent runs of experiments.

### 6.7.3 Parameter Setting

For all the kernel methods, I use the RBF kernel, which is defined as

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\delta^2}) \tag{6.16}$$

The hyperparameter $\delta$ often significantly influences the performance of kernel methods. With the help of constraints, the $\delta$ value is determined by a simple

grid search. For a given $\delta$, only the kernel null space projection is performed, then the projected data are clustered. Since the kernel null space projection guarantees that all must-linked data points will be trivially clustered together, the $\delta$ value that achieves the maximal clustering accuracy on cannot-link constraints is picked. Empirical results show that this method works very well even with a few pairs of constraints. The $\delta$ values chosen for each data set are listed in Table 6.1. The number of nearest neighbors used in constructing the adjacency and disjoint matrices is set to 5 and is kept the same for all the methods and all the data sets.

### 6.7.4 Fixed Subspace Dimensions

This experiment tests the dimension reduction performance on data sets with moderate sizes. The purpose is to learn the best projection direction by using all the available data and evaluate the performance. For each data set and each cluster, I run the experiments by alternatively generating 5 and 20 random pairs of must-link and cannot-link constraints each based on class labels. This end up with $2 \times k \times 5(20)$ pairs of constraints in total for each data set, where $k$ is the number of clusters. For easy reference, I refer to them as "5(20) pairs" of constraints hereafter. The subspace dimension is fixed to be half of the original dimension. Table 6.2 shows the evaluation results. On 5 out of 7 data sets, DSP achieves the best F-scores. For the remaining 2 data sets, DSP still shows satisfactory F-scores. Most importantly, when the number of constraints is small (i.e. the 5 pairs

Table 6.2: F-score on half-size feature spaces

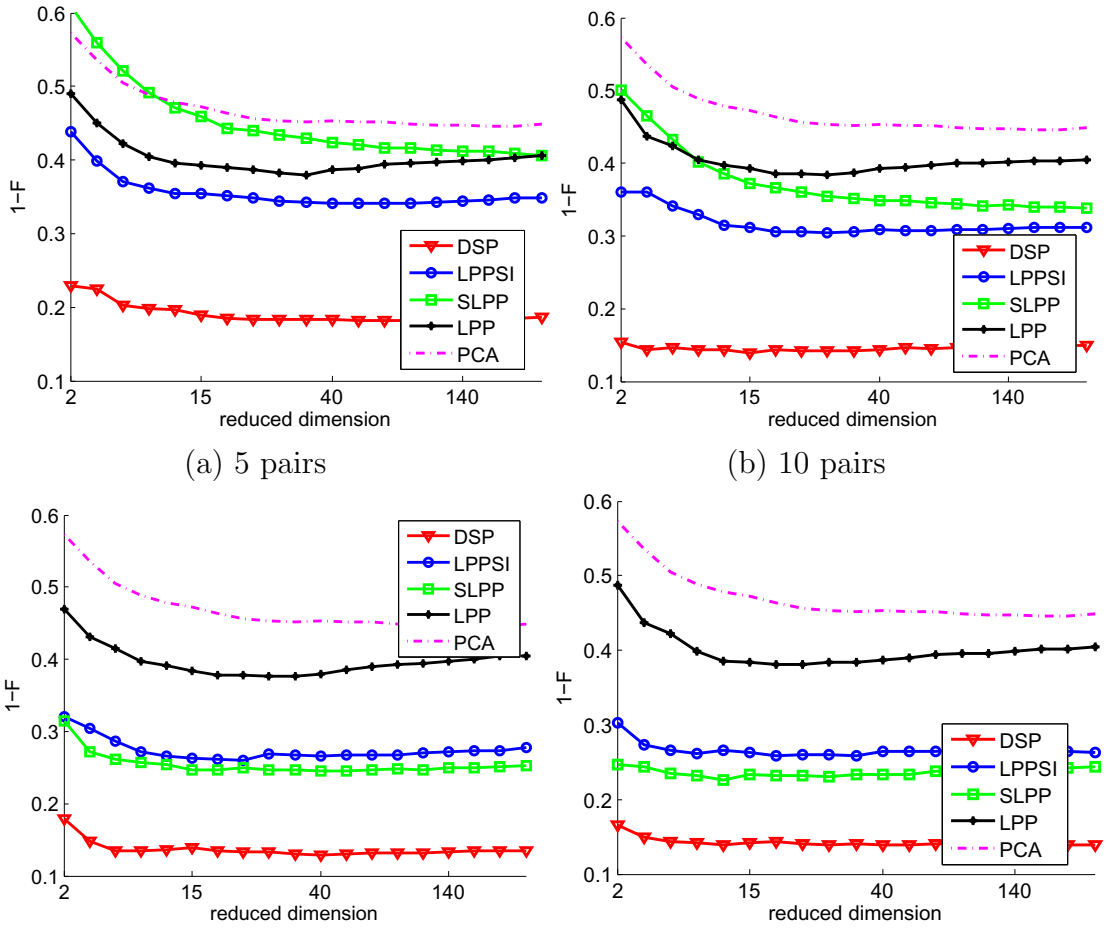
| | unsupervised | | 20 pairs | | | 5 pairs | | |
|---|---|---|---|---|---|---|---|---|
| | PCA | LPP | SLPP | LPPSI | DSP | SLPP | LPPSI | DSP |
| wine | 0.9415 | 0.9541 | 0.9563 | 0.8198 | **0.9588** | 0.5962 | 0.7381 | **0.9322** |
| vehicle | 0.3070 | 0.3383 | 0.6024 | 0.4092 | **0.6042** | 0.3417 | 0.3306 | **0.3604** |
| iris | 0.8112 | 0.7716 | 0.8920 | 0.6982 | **0.9498** | 0.8471 | 0.6244 | **0.9405** |
| balance | 0.5075 | 0.4754 | 0.5789 | 0.5800 | **0.6068** | 0.5749 | **0.5845** | 0.5693 |
| ionoshpere | 0.6050 | 0.6050 | 0.7061 | 0.6205 | **0.7211** | 0.6108 | 0.5992 | **0.7145** |
| glass | 0.3950 | 0.3903 | **0.4032** | 0.4023 | 0.3833 | 0.3849 | 0.3058 | **0.4131** |
| breast | 0.9307 | 0.9307 | 0.9027 | **0.9352** | 0.9202 | 0.7478 | **0.9292** | 0.9288 |

case), the performance of DSP is still robust and is better than or similar to the performances of the two unsupervised method PCA and LPP. This means that DSP does not suffer from overfitting, unlike competing methods.

### 6.7.5 Various Subspace Dimensions

This experiment evaluates the dimension reduction techniques for various subspace dimensions on the 3D object recognition task and the handwritten digit recognition task. Results on the COIL-20 database for 3D object recognition and the Multiple Features data set for handwritten digit recognition are shown in Figures 6.4 and 6.5 respectively. For each data set 5/10/20/30 pairs of constraints per cluster are generated following the procedure introduced in last experiment. The reduced dimensions range from 2 to 200. DSP significantly outperforms other dimension reduction techniques for both data sets under all experiment settings. The stable performance of DSP given a few constraints and very low subspace dimensionality is particularly impressive. It is interesting to notice that although LPPSI and SLPP perform well for the COIL-20 data set, their performances on the digit data

set are worse than the unsupervised LPP for low dimensions and small number of constraint pairs. This effect could be the result of overfitting due to few training data.



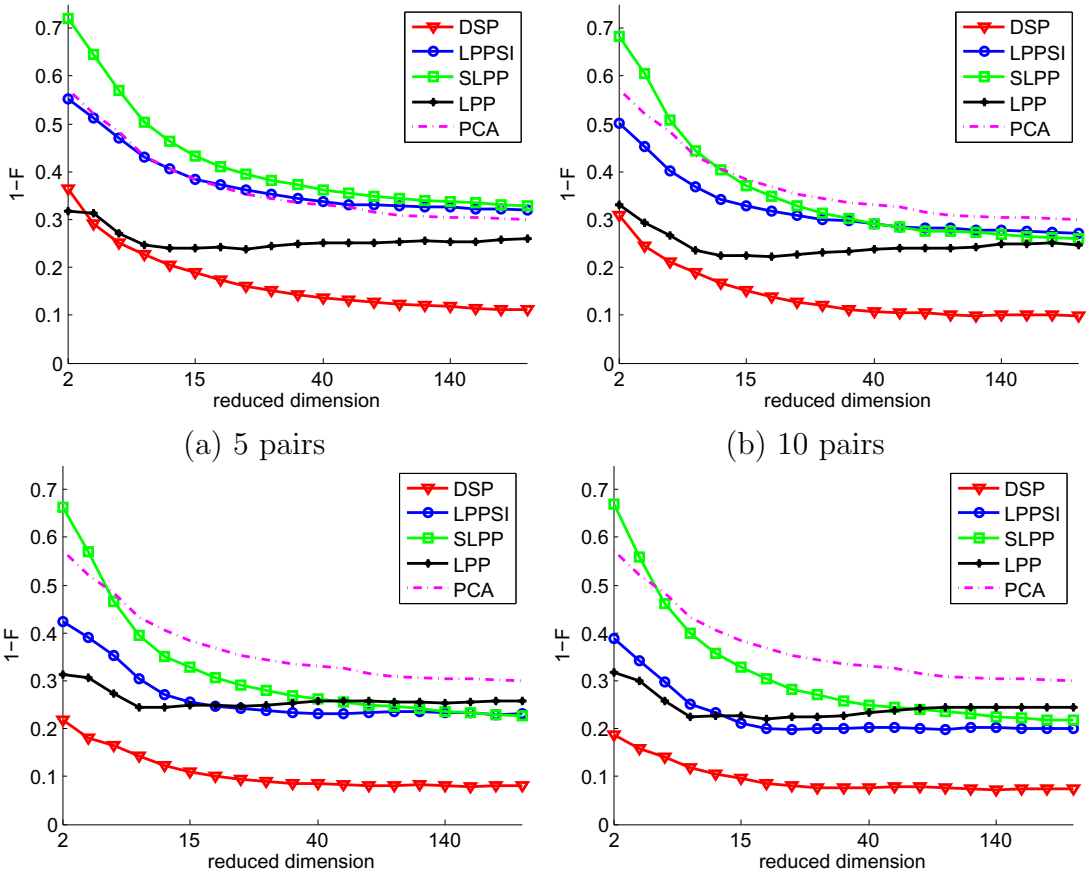

(a) 5 pairs

(b) 10 pairs

(c) 20 pairs

(d) 30 pairs

Figure 6.4: Error Rate vs. Reduced Dimensions for 3D object recognition




(a) 5 pairs (b) 10 pairs

(c) 20 pairs (d) 30 pairs

Figure 6.5: Error Rate vs. Reduced Dimensions for handwritten digit recognition

### 6.7.6 Generalization

This experiment evaluates how well DSP handles new data points on four large scale data sets. For each data set, I do 5-fold cross validation. Four folds of data are used for training. The training part includes generating 20 pairs of constraints and learning the best subspace embedding. Then the one fold testing data points are projected to the learned subspace for further clustering evaluation. Table 6.3 shows the generalization performance, compared to the result of clustering

Table 6.3: F-score for Generalization ($r$: subspace dimensionality)

| | full feature | DSP-generalize($r$) |
|---|---|---|
| Multiple Features | 0.7101 | 0.9459(20) |
| isolet | 0.5311 | 0.4740(20) |
| Pendigit | 0.5502 | 0.5873(5) |
| COIL-20 | 0.5732 | 0.7872(20) |

testing data without dimension reduction. Because the subspace dimensions are significantly smaller than the dimensions of the original full input space, clustering in the subspace will most of the time sacrifice accuracy for efficiency. With the help of constraints, for three data sets, the clustering accuracy after DSP reduction is in fact being improved. This indicates that DSP is effective in exploiting constraints and generalizing to new data points.

## 6.8 Appendix

Let us prove that in the null space of $\mathbf{M}$, every pair of must-linked data points collapse to a single point, and thus the must-link constraints are maximally satisfied.

**Proof 3.** *Let $(\phi(\mathbf{x}_i), \phi(\mathbf{x}'_i))$ be the $i$-th pair of must-link data points in the kernel space $\mathcal{H}$. For any data point $\phi(\mathbf{x}) \in \mathcal{H}$, its embedding in the null space of $\mathbf{M}$ is given by*

$$\hat{\phi}(\mathbf{x}) = \mathbf{P}\phi(\mathbf{x}) \tag{6.17}$$

*we then have*

$$\hat{\phi}(\mathbf{x}_i) - \hat{\phi}(\mathbf{x}'_i) = \mathbf{P}(\phi(\mathbf{x}_i) - \phi(\mathbf{x}'_i))$$

$$\begin{aligned} &= (\mathbf{I} - \mathbf{U})(\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) \\ &= (\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) - \mathbf{U}(\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) \\ &= (\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) - (\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) \\ &= 0 \end{aligned} \tag{6.18}$$

*The identity* $\mathbf{U}(\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i)) = (\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i))$ *follows from the fact that* $(\phi(\mathbf{x}_i) - \phi(\mathbf{x'}_i))$ *is in the row space of* $\mathbf{M}$. *Since* $\mathbf{P}$ *is not null, we get*

$$\hat{\phi}(\mathbf{x}_i) = \hat{\phi}(\mathbf{x'}_i) \tag{6.19}$$

*Thus the two points are mapped to the same point.*

Chapter 7

# Future Work and Conclusion

## 7.1 Future Work Directions

High-dimensional data are prevalent in real-world applications in various domains. Learning from high-dimensional data efficiently and effectively is a constant need both in research and in practice. Side information other than labeled data can improve learning with much less human effort. The study of semi-supervised learning on exploring side information to improve the analysis of high-dimensional data is a research topic that deserves more efforts. Besides the issues that are covered in the thesis, this section describes several directions for future work that are of key interest.

**Fuzzy Side Information** As introduced in Section 1.2, side information can take various forms. No matter which form of side information is under consideration, existing work on semi-supervised clustering assumes the information

is clear and straightforward. For example, either the two data items **a** and **b** belong to a same cluster or not. However, fuzzy side information is possible and is more realistic in several scenarios. For example, suppose users provide side information to direct a clustering process. It may be difficult for a user to tell for sure whether two data items belong to a same cluster or not. On the contrary, it is a lot easier for a user to describe that item **a** is more similar to item **b** than to item **c**. Mathematically, this leads to the side information expressed as $dist(\mathbf{a}, \mathbf{b}) < dist(\mathbf{a}, \mathbf{c})$. Or the user can state that items **a** and **b** are more similar to each other than items **c** and **d** are to each other. This information is expressed as $dist(\mathbf{a}, \mathbf{b}) < dist(\mathbf{c}, \mathbf{d})$. Moreover, given two clustering solutions $X$ and $Y$, a user may point out that solution $X$ is better than solution $Y$, although neither is a perfect solution. The user's judgement provides information that can be explored to improve a clustering scheme. Because such side information is not straightforward, I call them fuzzy side information. To exploit fuzzy side information will be a challenging yet worthy direction to go.

**Active Learning** Active learning is a closely related field to semi-supervised learning. Given a data set to be clustered, the number of possible pairwise constraints is huge. Presenting a large amount of pairwise queries to users is not realistic. In active learning, the learner picks the constraint query that will improve learning most, and asks an oracle or user for side informa-

tion. Among all the pairwise queries, which subset of queries can improve the learning accuracy most? The answer to this question varies according to clustering approaches. In general, the goal of introducing active learning in semi-supervised clustering is to let the learning system adaptively and optimally select the best queries about constraints. Existing work has studied the active learning problem for distance-metric-learning-based methods. For dimensionality-reduction-based semi-supervised clustering methods, the problem of actively selecting the most informative queries is not well studied.

**Automatic Side Information Generation** Deriving side information automatically from domain knowledge is a key factor that can influence the applicability of semi-supervised learning techniques to real-world applications. In this thesis, I have studied this problem for the application of networked document clustering. The proposed method can be easily adjusted to fit other application domains. For example, in the task of segmenting a video according to human faces or objects appeared in the video, faces/objects that appear in roughly the same position in consecutive fames can be considered as "strongly connected" and constraints can be generated accordingly. However, for many other domains, automatically deriving constraints requires in-depth understanding of domain knowledge, as well as designing schemes that are adaptive to the domain under consideration. For example, for the record linkage problem arising in many database applications, "blocking" can

be performed to generate side information [YLKG07]. Blocking is a cheap and fast preprocessing step that partitions a large list of records into disjoint blocks, so that the more expensive detailed comparisons of records are performed only within each block. Thus, record from different blocks can be considered as candidates for cannot-link constraints.

## 7.2 Conclusion

Research presented in this thesis has focused on semi-supervised clustering of high-dimensional data with sparse features based on dimension reduction given pairwise supervision. I have shown that by exploiting pairwise must-link and cannot-link constraints, high-dimensional data can be embedded into a much reduced-dimension subspace where the clustering structure of data is more evident. The low-dimensional embedding thus leads to more efficient and more effective clustering solutions. The better quality clusters are essential for exploratory data analysis or for the subsequent supervised classification tasks.

Novel semi-supervised clustering approaches introduced in this thesis are based on dimension reduction techniques. Compared to traditional semi-supervised clustering techniques based on distance metric learning, dimension-reduction-based methods have the advantage of being efficient and effective in handling high-dimensional data. The number of parameters needs to be learned in distance learning is quadratic to the number of features. Thus, when dealing with high-

dimensional data, a metric learning based technique cannot learn a robust and informative distance given limited constraints. Besides, distance learning is usually reduced to solving a convex optimization problem with gradient descent and iterative projection, and often suffers from large computation cost. On the contrary, dimension-reduction-based techniques use limited constraints to find a better data representation, and the optimal low-dimensional embedding is found by exploiting both a small set of constraints and the structure of a large amount of unlabeled data. Thus, dimension-reduction-based techniques proposed in the thesis yield robust clustering solutions even with very limited side information. Moreover, most dimension-reduction-based techniques boil down to a general eigenvalue problem to which well-studied solutions exist and can be solved efficiently.

First, I study how to explore both constraints and unlabeled data in semi-supervised dimension reduction efficiently. Although constraints encode the desired clustering criteria provided by user or derived from domain knowledge, a clustering scheme that is entirely based on constraints may lose the important structure information of the data set. Instead, I find that constraints define an approximate-clustering structure on data, and the goal of semi-supervised dimension reduction is to embed high-dimensional data in a low-dimensional subspace such that the approximate-clustering structure is not only preserved but also enhanced. When applying a clustering scheme to the embedded data, better quality clusters can be expected since the clustering structure is more evident after di-

mension reduction. This idea leads to the scheme of semi-supervised clustering by approximate-structure-preserving dimension reduction (ASP). ASP performs robustly given limited constraints, and significantly improves clustering accuracy as the number of constraints increases. For high-dimensional data whose feature space dimension is larger than the number of items, i.e., most text data sets, the reduced QR factorization technique can be adopted for fast ASP reduction. Otherwise, the reduced SVD factorization technique can be adopted to project data to a desired dimension.

I then study the noisy constraints issue involved in real applications of semi-supervised clustering. In many real-world applications, pairwise constraints can be automatically derived from domain knowledge. However, such constraints are inevitably noisy. Most existing semi-supervised clustering techniques are designed for well-defined noise-free constraints, and do not perform well given noisy constraints. I focus my work on the networked document clustering domain, although the proposed technique is equally applicable to other domains.

In the first part of the work, I study how to automatically derive high-quality pairwise constraints from domain knowledge. In particular, constraints are extracted from the link structure of networked documents as complement to text-based content information. A local and a global link analysis methods are proposed to extract robust link information from noisy and sparse link graphs. These methods analyze link graphs from a local and a global view respectively. Then I bridge

the disconnection between content and links by searching for an optimal subspace data representation, where the search space is constrained by both content similarity and link structure similarity. The propose content & structure constrained (Costco) feature projection method couples content and link structure in a unified objective function, and hence avoids heuristic combination of the two information sources. Besides, the method does not rely on the availability of dense link structure and is robust to noisy links, which suits it well for real-world networked data. Moreover, the method is very simple to implement, so can be used for exploratory data analysis before any complicated in-depth analysis.

I further study the non-linear separability problem in dimensionality-reduction-based semi-supervised clustering. Dimension reduction can be achieved by either linear or non-linear transformations. Linear transformations are simple to compute and are analytically tractable. In general, linear transformations perform well. However, when sever non-linearity is involved in data, and clusters are not linearly separable, non-linear transformations, such as kernel-based techniques are usually adopted. However, non-linear methods have many drawbacks as have been introduced in Section 6.3. To this end, I propose semi-supervised dimension reduction approach that benefits from both linear and non-linear transformations. In particular, I propose a novel semi-supervised dimension reduction technique based on dual subspace projections (DSP) in both the kernel space and the input space. Projections in the two spaces interact and data are embedded in an optimal low-

dimensional subspace where the intrinsic structure of data is more evident, and thus eases the subsequent data analysis. Significant improvement in clustering quality is achieved after the DSP dimension reduction with only a few constraints.

Overall, the work presented in this thesis contributes methods leading to state-of-the-art performance on semi-supervised clustering tasks. This research demonstrates the power of using dimension reduction techniques and side information in the form of pairwise constraints in learning from high-dimensional data with sparse features. I hope that the thesis work will motivate further research in dimensionality reductions, semi-supervised clustering, and high-dimensional data analysis, and encourage such techniques in various applications where analyzing high-dimensional data is essential.

# Bibliography


[AF95] David Aldous and James Allen Fill. Reversible markov chains and random walks on graphs-chapter 9: A second look at general markov chains, 1995.

[ALV08] Senjian An, Wanquan Liu, and Svetha Venkatesh. Exploiting side information in locality preserving projection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[AS06] Ralitsa Angelova and Stefan Siersdorfer. A neighborhood-based approach for clustering of linked document collections. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 778–779, 2006.

[BBM02] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Semi-supervised clustering by seeding. In *International Conference on Machine Learning (ICML)*, pages 27–34, 2002.

[BBM04a] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. A probabilistic framework for semi-supervised clustering. In *ACM International conference on Knowledge discovery and data mining (KDD)*, pages 59–68, 2004.

[BBM04b] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *International conference on Machine learning (ICML)*, 2004.

[BDJ99] Michael W. Berry, Zlatko Drmac, and Elizabeth R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41(2):335–362, 1999.

[BEG06] Levent Bolelli, Seyda Ertekin, and C. Lee Giles. Clustering scientific literature using sparse citation graph analysis. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, pages 30–41, 2006.

[Bel61] Richard E. Bellman. *Adaptive control processes - A guided tour*. Princeton University Press, Princeton, New Jersey, U.S.A., 1961.

[BHHSW03] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *International Conference on Machine Learning (ICML)*, pages 11–18, 2003.

[BLRR04] Avrim Blum, John Lafferty, Mugizi Robert Rwebangira, and Rajashekar Reddy. Semi-supervised learning using randomized mincuts. In *International conference on Machine learning (ICML)*, page 13, 2004.

[BN02] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2002.

[BNS06] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[Bol98] Daniel Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998.

[BP98] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *International World Wide Web Conference (WWW)*, 1998.

[CDI98] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *ACM's Special Interest Group on Management Of Data (SIGMOD)*, pages 307–318, 1998.

[CH00] David A. Cohn and Thomas Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Advances in Neural Information Processing Systems (NIPS)*, pages 430–436, 2000.

[Chu97] Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[CM02] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, August 2002.

[CMM03] Csaba Czirjek, Sean Marlow, and Noel Murphy. Face detection and clustering for video indexing applications. In *in Proceedings of Advanced Concepts for Intelligent Vision Systems*, pages 2–5, 2003.

[CS96] Peter Cheeseman and John Stutz. *Bayesian Classification (AutoClass): Theory and Results*, chapter 6, pages 62–83. 1996.

[CSZ93] Pak K. Chan, Martine D. F. Schlag, and Jason Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *International Conference on Design Automation (DAC)*, pages 749–754, 1993.

[CVJK08] Hakan Cevikalp, Jakob Verbeek, Frédéric Jurie, and Alexander Kläser". Semi-supervised dimensionality reduction using pairwise equivalence constraints. In *International Conference on Computer Vision Theory and Applications*, pages 489–496, 2008.

[DEYG+02] Shlomo Dubnov, Ran El-Yaniv, Yoram Gdalyahu, Elad Schneidman, Naftali Tishby, and Golan Yona. A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles. *Machine Learning*, 47(1):35–61, 2002.

[DGK04] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *ACM International conference on Knowledge discovery and data mining (KDD)*, pages 551–556, 2004.

[DHZ+01] Chris H. Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *IEEE International Conference on Data Mining (ICDM)*, pages 107–114, 2001.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[DM01] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):143–175, 2001.

[Don00] D. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. 2000.

[DR05] Ian Davidson and S. S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. In *SIAM International Conference on Data Mining (SDM)*, 2005.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jrg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM International conference on Knowledge discovery and data mining (KDD)*, pages 226–231, 1996.

[ETY+07] Ergin Elmacioglu, Yee Fan Tan, Su Yan, Min yen Kan, and Dongwon Lee. Psnus: Web people name disambiguation by simple clustering with rich features, 2007.

[For65] E. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–780, 1965.

[Fri88] J. H. Friedman. Regularized discriminant analysis. In *Journal of the American Statistical Association*, pages 165–175, 1988.

[GB04] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[GEJD07] Rong Ge, Martin Ester, Wen Jin, and Ian Davidson. Constraint-driven clustering. In *International conference on Knowledge discovery and data mining (KDD)*, pages 320 – 329, 2007.

[GFKT03] Lisa Getoor, Nir Friedman, Daphne Koller, and Benjamin Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research*, 3:679–707, 2003.

[GJW82] M. R. Garey, David S. Johnson, and Hans S. Witsenhausen. The complexity of the generalized lloyd - max problem. *IEEE Transactions on Information Theory*, 28(2):255–, 1982.

[GL89] G. H. Golub and C. F. Van Loan. *Matrix Computations*. JohnsHopkinsPress, second edition, 1989.

[HBhW04] Tomer Hertz, Aharon Bar-hillel, and Daphna Weinshall. Boosting margin based distance functions for clustering. In *International Conference on Machine Learning (ICML)*, pages 393–400, 2004.

[Hen05] Monika Henzinger. Hyperlink analysis on the world wide web. In *HYPERTEXT*, pages 1–3, 2005.

[HJP03] Peg Howland, Moongu Jeon, and Haesun Park. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 25(1):165–179, 2003.

[HLLM06] Steven C. H. Hoi, Wei Liu, Michael R. Lyu, and Wei-Ying Ma. Learning distance metrics with contextual constraints for image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2072–2078, 2006.

[HN03] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Educational Psychology*, 24:417–441, 1933.

[HZG05] Hui Han, Hongyuan Zha, and C. Lee Giles. Name disambiguation in author citations using a k-way spectral clustering method. In *The Joint Conference on Digital Libraries (JCDL)*, pages 334–343, 2005.

[HZHDDS02] Xiaofeng He, Hongyuan Zha, Chris H.Q. Ding, and Horst D. Simon. Web document clustering using hyperlink structures. *Computational Statistics & Data Analysis*, 41(1):19–45, 2002.

[JMF99] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Survey*, 31(3):264–323, September 1999.

[JPR01] M Jeon, H. Park, and J.B. Rosen. Dimension reduction based on centroids and least squares for efficient processing of text data. In *SIAM International Conference on Data Mining (SDM)*, 2001.

[JX06] Xiang Ji and Wei Xu. Document clustering with prior knowledge. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 405–412, 2006.

[KKM02] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *International Conference on Machine Learning (ICML)*, pages 307–314, 2002.

[Kle99] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[LJJ07] Yi Liu, Rong Jin, and Anil K. Jain. Boostcluster: boosting clustering by pairwise constraints. In *International conference on Knowledge discovery and data mining (KDD)*, pages 450–459, 2007.

[Llo82] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.

[Mac67a] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[Mac67b] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[Men04] Filippo Menczer. Lexical and semantic clustering by web links. *Journal of the American Society for Information Science and Technology (JASIST)*, 55(14):1261–1269, 2004.

[MN98] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification, 1998.

[MRW+99] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Signal Processing Society Workshop*, pages 41–48, 1999.

[MS99] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

[MS00] Dharmendra S. Modha and W. Scott Spangler. Clustering hypertext with applications to web searching. In *HYPERTEXT*, pages 143–152, 2000.

[NAJ03] Jennifer Neville, Micah Adler, and David Jensen. Clustering relational data using attribute and link information. In *the Text Mining and Link Analysis Workshop of International Joint Conferences on Artificial Intelligence (IJCAI)*, 2003.

[NG00] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 86–93, 2000.

[OML00] Hyo-Jung Oh, Sung Hyon Myaeng, and Mann-Ho Lee. A practical hypertext catergorization method using links and incrementally available class information. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 264–271, 2000.

[Pea01] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901.

[PT03] Han Woo Park and Mike Thelwall. Hyperlink analyses of the world wide web: A review. *Journal of Computer-Mediated Communication*, 8(4), 2003.

[RS00] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[SEK03] Michael Steinbach, Levent Ertz, and Vipin Kumar. The challenges of clustering high-dimensional data. In *In New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recognition*, 2003.

[SM86] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[SM97a] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, page 731, 1997.

[SM97b] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, page 731, 1997.

[SRS03] Lawrence K. Saul, Sam T. Roweis, and Yoram Singer. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[SSGM00] Alexander Strehl, Er Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. In *AAAI Workshop on Artificial Intelligence for Web Search*, pages 58–64, 2000.

[SSM97] Bernhard Schölkopf, Alex J. Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks (ICANN)*, pages 583–588, 1997.

[SSM98] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[STC04] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[SWY97] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. pages 273–280, 1997.

[TAK02] Benjamin Taskar, Pieter Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 485–492, 2002.

[Tik63] A. N. Tikhonov. Regularization of incorrectly posed problems. In *Soviet Math (English Translation)*, volume 4, 1963.

[TmCK05] Ivor W. Tsang, Pak ming Cheung, and James T. Kwok. Kernel relevant component analysis for distance metric learning. In *In IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 954–959, 2005.

[TPM09] Oncel Tuzel, Fatih Porikli, and Peter Meer. Kernel methods for weakly supervised mean shift clustering. In *International Conference on Computer Vision (ICCV)*, pages 59–68, 2009.

[TXZW07] Wei Tang, Hui Xiong, Shi Zhong, and Jie Wu. Enhancing semi-supervised clustering: a feature projection perspective. In *ACM International conference on Knowledge discovery and data mining (KDD)*, pages 707–716, 2007.

[Vap98] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998.

[Ver03] Michel Verleysen. Learning high-dimensional data. In *Limitations and Future Trends in Neural Computation*, pages 141–162, 2003.

[vR79] C. J. van Rijsbergen. *Information Retrieval*. 2 edition, 1979.

[WC00] Kiri Wagstaff and Claire Cardie. Clustering with instance-level constraints. In *International Conference on Machine Learning (ICML)*, pages 1103–1110, 2000.

[WCRS01] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning (ICML)*, pages 577–584, 2001.

[Web02] Andrew Webb. A kernel approach to metric multidimensional scaling. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 613–629, 2002.

[WK02] Yitong Wang and Masaru Kitsuregawa. Evaluating contents-link coupled web page clustering for web search results. In *ACM Conference on Information and Knowledge Management (CIKM)*, pages 499–506, 2002.

[WZL07] Fei Wang, Changshui Zhang, and Tao Li. Regularized clustering for documents. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 95–102, 2007.

[XNJR03] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems (NIPS)*, pages 505–512, 2003.

[Yar95] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Annual Meeting on Association for Computational Linguistics*, pages 189–196, 1995.

[YD06] Bojun Yan and Carlotta Domeniconi. Subspace metric ensembles for semi-supervised clustering of high dimensional data. In *European Conference on Machine Learning (ECML)*, pages 509–520, 2006.

[YL07] Su Yan and Dongwon Lee. Toward alternative measures for ranking venues: a case of database research community. In *The Joint Conference on Digital Libraries (JCDL)*, pages 235–244, 2007.

[YLKG07] Su Yan, Dongwon Lee, Min-Yen Kan, and C. Lee Giles. Adaptive sorted neighborhood methods for efficient record linkage. In *The Joint Conference on Digital Libraries (JCDL)*, pages 185–194, 2007.

[YS04] Stella X. Yu and Jianbo Shi. Segmentation given partial grouping constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):173–183, 2004.

[YWLG09] Su Yan, Hai Wang, Dongwon Lee, and C. Lee Giles. Pairwise constrained clustering for sparse and high dimensional feature spaces. In *The Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 620–627, 2009.

[ZG09] Xiaojin Zhu and Andrew B. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan & Claypool Publishers, 2009.

[ZGL03] Xiaojin Zhu Zhuxj, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *International conference on Machine learning (ICML)*, pages 912–919, 2003.

[Zho05] Shi Zhong. 2005 special issue: Efficient streaming text clustering. *Neural Netwwork.*, 18(5-6):790–798, 2005.

[ZHT05] Liang Zhao, Nagamochi Hiroshi, and Ibaraki Toshihide. Greedy splitting algorithms for approximating multiway partition problems. *Mathematical Programming manuscript*, 102(1):167–183, 2005.

[ZZC07] Daoqiang Zhang, Zhi-Hua Zhou, and Songcan Chen. Semi-supervised dimensionality reduction. In *SIAM International Conference on Data Mining (SDM)*, 2007.

# Vita

## Su Yan

Su Yan is a Ph.D. student in the College of Information Sciences and Technology of Pennsylvania State University. Her Ph.D. advisor is Professor Dongwon Lee. Her research interests are in the area of machine learning and data mining. Before this, she received her MS degree and Bachelor degree from the Electrical Engineering Department of Jilin University, China. During her Ph.D. study, she had worked as intern at IBM Silicon Valley Lab for two summers, and at the Mitsubishi Electric Research Laboratories for one summer and one fall semester.

Permanent Address : 4th door, #401
Lixin St. #27,
Changchun, Jilin
P.R. China, 130021