This is to certify that the
dissertation entitled

Semi-supervised Learning with Side Information:
Graph-based Approaches

presented by

Yi Liu

has been accepted towards fulfillment
of the requirements for the

Ph.D. degree in Computer Science and Engineering

Major Professor's Signature

12/06/2007.

Date

SEMI-SUPERVISED LEARNING WITH SIDE INFORMATION: GRAPH-BASED APPROACHES

By

Yi Liu

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science and Engineering

2007

ABSTRACT

SEMI-SUPERVISED LEARNING WITH SIDE INFORMATION: GRAPH-BASED APPROACHES

By

Yi Liu

In many real-world learning tasks, data examples are known not only by their input patterns, but also in other forms that are often refered to as "side information". Side information provides additional knowledge about the data, leaving hope for machine learning algorithms to gain more insight into the structure of data and thus perform better. However, the usual incomplete, sparse, and noisy nature of side information also poses challenges. This dissertation will present research work in semi-supervised learning with side information.

Semi-supervised learning uses both labeled and unlabeled data in training. In this work, we follow the graph-based approaches that are able to explore the underlying structure of data by constructing a graph over data examples. Taking such an advantage of graph representation for semi-supervised learning, we propose to construct a graph over all labeled and unlabeled data and further incorporate side information into the graph. To effectively utilize side information against its incompleteness, sparseness, and noise, this work adopts a common theme in formalizing graph-based learning models, i.e., enforcing consistency over the graph. More specifically, consistency is maximized among data input patterns, supervised information (if any), side information, and predictions on the unlabeled data. Optimization approaches are taken to carry out the consistency enforcement, with objective functions that collect consistency measurement defined on all possible data pairs.

Solutions to three generic learning tasks are presented to illustrate the proposed method of utilizing side information in a semi-supervised learning setting. Specifically,

we study multi-label learning with class correlation information, to meet the challenge of a large number of classes and a small training set. To improve classification with link information, we propose a boosting framework that is able to improve the classification accuracy of any given supervised classifiers. Another boosting framework is also designed to boost any given clustering algorithm, with the help of pairwise constraints over the data.

In addition, two applications in the area of information retrieval will be discussed. In the first application, we develop a maximum coherence framework to tackle the difficulty of query translation disambiguation in cross-language information retrieval, with a bilingual dictionary as side information. The proposed framework will also be explained as two-way graph partitioning. The second application is automatic extraction of question-answer pairs from Web FAQs, where the side information comes from human knowledge on the presentation regularity on Web FAQ pages. Correlated label propagations over a graph constructed for each FAQ page is shown to be an interpretation of the corresponding model.

All the semi-supervised learning models proposed for the tasks and applications demonstrate the effectiveness of the consistency enforcement theme in exploiting side information for semi-supervised learning. Analysis shows that the proposed models are robust against the incompleteness, sparseness, and noise of side information, and retain the power of utilizing both labeled and unlabeled data for training as semi-supervised learning.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Learning with inadequate amount of, or no, supervised information poses a challenge for machine learning research. In practical domains, though supervised information may be hard to collect, it is often the case that some certain partial supervised information is available. This naturally leads to a question: *can we learn better, if more knowledge on the data is gained?*

Answering this question leads to the research topic of semi-supervised learning, which has gained significant attention in recent years.

Semi-supervised learning, in a strict sense, refers to the family of classification techniques in machine learning that makes use of both labeled and unlabeled data. In a typical semi-supervised learning setting, the amount of labeled data is small while the amount of unlabeled data is large. As its name suggests, semi-supervised learning lies in the continuum of two ends in the spectrum of machine learning – strictly supervised learning (with completely labeled data), and strictly unsupervised learning (without any labeled data). Usually, some *partial* supervised information is available in semi-supervised learning, which typically includes the observation of all unlabeled data that is going to be classified into predefined categories, possibly as well as information in other forms that do not directly provides class labels.

Data clustering is a typical unsupervised learning problem, where no supervised information is available. The word "semi-supervised" has also been introduced to data

clustering, i.e. "semi-supervised clustering", when *partial* supervised information is available for clustering purpose.

Regardless of how the terminology evolves, the essence of semi-supervised learning is the utilization of any knowledge gained in addition to the supervised information. The particular interest we will show in this thesis, is the kind of additional knowledge beyond the input patterns of the data. By input patterns, we refer to the observation of "feature representations" (or also known as "attribute values" in literature) on all the data. This leads to the research issue that centers this thesis work, i.e.

How can we improve semi-supervised learning, if we know anything more than the input patterns of labeled data (if any) and unlabeled data?

The knowledge about data that is not the input patterns will be referred to as "side information" in this thesis. Also, among all types of semi-supervised learning work, we will focus on graph-based approaches.

In this chapter, we will first review graph-based semi-supervised learning, then motivates the use of side information in semi-supervised learning, and briefly overview the entire thesis work. Detailed discussions on each piece of work will be left to the rest chapters.

1.1 Graph-based Semi-supervised Learning

1.1.1 Mathematical Definition of Graph

From a mathematical point of view, a graph is a collection of points and lines connecting some (possibly empty) subset of them [151]. The points of a graph are most commonly known as graph vertices, but may also be called "nodes"or simply "points". Similarly, the lines connecting the vertices of a graph are most commonly known as graph edges, but may also be called "links", "arcs" or simply "lines". In an *undirected graph*, edges are not directional, i.e., a line from point A to point B is not distin-

guished from a line from point B to point A. However, the two directions are distinct in a *directed graph* (or *digraph* for short). On many occasions, a weight (usually positive) will be associated with each edge, indicating the strength of the relationship within the corresponding vertex pair.

Formally, we can denote a graph by $G(V, E)$, where V is the vertex set and E is the edge set. For a finite graph G with n vertices, the *adjacency matrix* is defined as a binary matrix $\mathbf{A} = \left[a_{i,j}\right]_{n\times n}$, with $a_{i,j} = 1$ denoting there is an edge between the i-th and the j-th vertices and $a_{i,j} = 0$ otherwise.

Introducing the weight to each edge in a graph will result in a *weighted graph*. In this case the adjacency matrix (or *weight matrix*) becomes $\mathbf{W} = \left[w_{i,j}\right]_{n\times n}$, with $w_{i,j} > 0$ indicating the edge weight between the i-th and the j-th vertices and $w_{i,j} = 0$ indicating no edge there. For an undirected graph, both the adjacency matrix and the weight matrix are symmetric.

An important concept centering in Spectral Graph Theory is the *graph Laplacian*, which has been widely used in graph-based semi-supervised learning models. With respect to a symmetric adjacency matrix $\mathbf{W}$, the graph Laplacian is defined as follows

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{1.1}$$

where $\mathbf{D}$ is a diagonal matrix $\mathbf{D} = diag(d_1, d_2, \cdots, d_n)$ with $d_i = \sum_{j\in V} w_{i,j}$. A normalized version of the matrices above is defined in some occasions

$$\tilde{\mathbf{W}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$$

and

$$\begin{aligned}\tilde{\mathbf{L}} &= \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} \\ &= \mathbf{I} - \tilde{\mathbf{W}}\end{aligned}$$

The graph Laplacian $\mathbf{L}$ (or $\tilde{\mathbf{L}}$) has the following properties

1. It is positive semi-definite, i.e., all the eigenvalues are non-negative;

2. Its minimum eigenvalue is always 0. For unnormalized graph Laplacian $\mathbf{L}$, the corresponding principal eigenvector is $\mathbf{e} = (1/\sqrt{n}, 1/\sqrt{n}, \ldots, 1/\sqrt{n})^\top$; for normalized graph Laplacian $\tilde{\mathbf{L}}$, the corresponding principal eigenvector is $\mathbf{D}^{\frac{1}{2}}\mathbf{e}$

The set of eigenvalues of the graph Laplacian $\mathbf{L}$ can be denoted by $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1}$, which is also called the *spectrum* of $\mathbf{L}$ (or the graph itself). Spectral Graph Theory tells us that the structure of a graph and its principal properties can be deduced from its spectrum. In particular, it has been shown that the eigenvalues are closely related to almost all major invariants of the graph, and link external properties together ([38, 37]). As will be shown later in the next subsection, graph Laplacian is clearly present in a number of graph-based learning models.

1.1.2 Graph-based Semi-supervised Learning Models

Semi-supervised learning deals with the use of both labeled and unlabeled data for training. The central idea behind nearly all semi-supervised learning algorithms is an assumption made on the data consistency that data examples close to each other or in the same structure are more likely to have the same label. Various semi-supervised learning approaches differ in the way to model the structure of data and attempts to propagate the label information from labeled examples to unlabeled ones. One approach is to construct a graph over all the labeled and unlabeled data: each vertex represents a data example; whenever a pair of data examples are close enough by some similarity measurement, an edge is constructed between them with a weight proportional to their similarity value. Taking this graph point of view, a number of graph-based semi-supervised learning model can have interpretation in graph languages, such as graph partitioning [25, 26, 87], random walk on the graph [134, 119, 70], ranking of nodes [119, 70, 92, 144], graph approximation [104, 105], and etc..

In the rest of this subsection, we will recapitulate a few typical graph-based semi-supervised learning models.

Graph Mincuts

The graph mincuts model extends the algorithm for finding the minimal cut in a graph to a transductive learning setup [25, 26]. The basic idea is searching for a partition of the graph which results in a minimum sum of weights of the edges being cut while agreeing with the labeled data. To enforce the consistency with the labeled data, a special weighting scheme is adopted to build a graph over all the labeled and unlabeled examples: for any pair of data examples belonging to different classes, the edge weight indicates their similarity; for any pair of data examples belonging to the same class, an *infinite* weight is assigned. In a binary case, the search for the minimum cut amounts to the following optimization problem

$$\begin{aligned} \min_{\{y_i|\mathbf{x}_i\in\mathbf{X}^U\}} \quad & \sum_{i,j} w_{i,j}(y_i - y_j)^2 \\ s.t. \quad & y_i = y_i^*, \forall \mathbf{x}_i \in \mathbf{x}^L \end{aligned}$$

where y_i^* indicates the known labels of the labeled data.

Note that the solution gives *binary* labels for the unlabeled data, which can also been proved to be optimal in another sense that it minimizes the leave-one-out cross-validation error of the nearest-neighbor algorithm applied to the entire dataset ([25]). To improve the robustness of the solution, a follow-up work ([26]) introduces random noise to edge weights and results in a solution with "soft" labeling.

Gaussian Random Fields and Harmonic Functions

Gaussian random fields and harmonic functions method ([173, 172, 174]) is motivated by the assumption that the label probability should vary smoothly over the entire graph. To enforce the label smoothness on the graph, a quadratic energy function is

proposed as follows (in a binary classification case)

$$\begin{aligned} E(\mathbf{f}) &= \frac{1}{2}\sum_{i,j} w_{i,j}(f_i - f_j)^2 \\ &= \mathbf{f}^\top \mathbf{L}\mathbf{f} \end{aligned}$$

where $\mathbf{f} = (f_1, f_2, \ldots, f_n)^\top$ is the label probability vector defined as

$$f_i = \begin{cases} \delta(y_i, 1) & x_i \text{ is labeled} \\ \Pr(y_i = 1|x_i) & x_i \text{ is unlabeled} \end{cases}$$

The energy defined above will be small when the label probability vector varies smoothly over the graph, which leads to the minimization of the energy function. The minimizer $\mathbf{f}^*$ makes the function *harmonic* in the sense that

$$(\mathbf{L}\mathbf{f}^*)_i = \begin{cases} y_i & x_i \text{ is labeled} \\ 0 & x_i \text{ is unlabeled} \end{cases}$$

This method is also related to Gaussian Random Field because the energy function could be used to form a Gaussian density function

$$p(\mathbf{f}) \propto \exp[-\beta E(\mathbf{f})]$$

where β is an "inverse temperature" parameter.

If we define $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ and further decompose it into four blocks

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{ll} & \mathbf{P}_{lu} \\ \mathbf{P}_{ul} & \mathbf{P}_{uu} \end{bmatrix}$$

where $\mathbf{P}_{ll}$ corresponds to the labeled data and $\mathbf{P}_{uu}$ corresponds to the unlabeled data, the final prediction is made in the following way

$$\mathbf{y}_u = (\mathbf{I} - \mathbf{P}_{uu})^{-1}\mathbf{P}_{ul}\mathbf{f}_l$$

To save the computation introduced by the matrix inverse, an extended work is proposed in [174], which essentially forms a backbone graph with super-nodes created by pre-clustering the examples.

Again the important role played by the graph Laplacian $\mathbf{L}$ is witnessed: the smoothness regularization on the graph and the harmonic nature of the energy function are both achieved through it, which suggests a close relationship to the Spectral Graph Theory ([173]).

Spectral Graph Transducer

Spectral Graph Transducer is another semi-supervised version of ratio-cut algorithm originally proposed for unsupervised learning ([87]). The objective function incorporates a quadratic penalty on labeled data in addition to minimizing the graph cut

$$\begin{aligned} \min_{\mathbf{f}} \quad & \mathbf{f}^\top \mathbf{L}\mathbf{f} + c(\mathbf{f}-\mathbf{r})^\top \mathbf{C}(\mathbf{f}-\mathbf{r}) \\ s.t. \quad & \mathbf{f}^\top \mathbf{e} = 0 \\ & \mathbf{f}^\top \mathbf{f} = n \end{aligned}$$

where the vector $\mathbf{f}$ is a label probability vector, and the vector $\mathbf{r}$ is defined as

$$r_i = \begin{cases} \hat{r}_+ & x_i \text{ is a positive example} \\ \hat{r}_- & x_i \text{ is a negative example} \\ 0 & x_i \text{ is unlabeled} \end{cases}$$

The matrix $\mathbf{C} = diag(c_1, c_2, \ldots, c_n)$ is a diagonal cost matrix allowing different misclassification cost for each data example. The trade-off between the graph cut value and the training error penalty is balanced through the constant c. It has been shown that solving the optimization problem in Spectral Graph Transducer also leads to a matrix eigen-decomposition problem ([87]).

Learning with Local and Global Consistency

The local and global consistency method ([166]) proposes the following optimization problem

$$\min_{\mathbf{F}} \quad \frac{1}{2}\sum_{i,j} w_{i,j} \left\| \frac{\mathbf{F}_i}{\sqrt{d_i}} - \frac{\mathbf{F}_j}{\sqrt{d_j}} \right\|^2 + \mu \sum_i ||\mathbf{F}_i - \mathbf{Y}_i||^2$$

where $\mathbf{F}$ is the label probability matrix with each column corresponding to a class, $d_i = \sum_{j \in V} w_{i,j}$ and $\mathbf{Y}$ is the label matrix with $y_{i,j} = 1$ if the i-th example is labeled as a member in the j-th class and $y_{i,j} = 0$ otherwise.

The first term in the object function addresses the smoothness constraint on the graph by the sum of local variations measured at each edge in an undirected graph; the second term penalize the inconsistency with the labeled data.

Local Laplacian Embedding

To learn the global manifold structure from the data, *local Laplacian embedding* methods propose to project the data from the original space to a dimension reduced space. In particular, let $\tilde{\mathbf{V}} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p]$ be a matrix composed by the p smallest eigenvector of the graph Laplacian for the nearest neighbor graph built over all labeled and unlabeled data. If we rewrite $\tilde{\mathbf{V}} = [\tilde{\mathbf{x}}_1^\top, \tilde{\mathbf{x}}_2^\top, \ldots, \tilde{\mathbf{x}}_n^\top]^\top$, $\tilde{\mathbf{x}}_i$ is the projected image of $\mathbf{x}_i$ in the dimension reduced space.

For classification purpose, in [18] a linear classifier is learned from the labeled data

$$\mathbf{a} \;=\; (\tilde{\mathbf{V}}_{LL}^\top \tilde{\mathbf{V}}_{LL})^{-1} \tilde{\mathbf{V}}_{LL}^\top \mathbf{y}^L$$

where $\tilde{\mathbf{V}}_{LL}$ is a sub-matrix of $\tilde{\mathbf{V}}$ that corresponds to the labeled examples. Then the prediction for the unlabeled data $\mathbf{x}_i$ is made by

$$y_i \;=\; \mathbf{a}^\top \tilde{\mathbf{x}}_i$$

Another manifold learning based semi-supervised learning algorithm extends the manifold ranking function ([169]). The prediction is as follows

$$\mathbf{y}^* = (\mathbf{I} - \alpha \mathbf{D}^{-1}\mathbf{W})^{-1}\mathbf{D}\mathbf{y}$$

where α is a constant that controls how much we rely on the labeled data, and $\mathbf{y}$ takes a ± 1 value if labeled and 0 if unlabeled.

Learning on Directed Graphs

Recently a semi-supervised learning method is proposed based on directed graphs [167], which extends the work in [166]. To address the data consistency, the undirected graph is expected to be smooth in the sense that nodes lying on a densely linked subgraph are likely to have the same label. To search for a good classifier which results in a labeling with such smoothness on the graph, a smoothness functional is proposed (for binary classification)

$$\Omega(\mathbf{f}) = \frac{1}{2}\sum_{i,j} \pi_i l_{i,j} \left(\frac{f_i}{\sqrt{\pi_i}} - \frac{f_j}{\sqrt{\pi_j}} \right)^2$$

where $\mathbf{f} = [f_1, f_2, \ldots, f_n]^\top$ is the label probability vector, $\mathbf{L} = [l_{i,j}]$ is the graph Laplacian over the entire data examples, and $\pi = [\pi_1, \pi_2, \ldots, \pi_n]^\top$ is the principal eigenvector of the graph Laplacian $\mathbf{L}$.

To find the labeling for the unlabeled data, an optimization problem is proposed with an objective function addressing both the data smoothness enforcement and the consistency with labeled data

$$\arg\min_{\mathbf{f}} \quad \Omega(\mathbf{f}) + \mu ||\mathbf{f} - \mathbf{y}||$$

where the component of $\mathbf{y}$ takes a ± 1 value if labeled and 0 if unlabeled, and $\mu > 0$ is a constant specifying the trade-off.

A slightly different version of the algorithm above is described in [168], where two sets of smoothness functionals are defined for the data examples: one for their "hub"

scores (which accounts for outgoing links) and one for their "authority" scores (which accounts for incoming links). To separate the two smoothness enforcement, directed graphs are transformed into bipartite graphs.

Spectral Clustering

Spectral clustering approaches view the problem of data clustering as a problem of graph partitioning. Taking 2-way graph partitioning as an example, to form two disjoint data sets A and B from a graph $G = (V, E)$, edges connecting these two parts should be removed. The degree of dissimilarity between the partitioned parts are captured by the notion of *cut*, which is defined as $Cut(A, B) = \sum_{v_i \in A, v_j \in B} w_{i,j}$. Generally speaking, a good partitioning should lead to a small cut value.

Addressing different balancing concerns, there are several variants of cut definition which lead to the optimal partitioning in different senses. To begin with, we define $S(A, B) = \sum_{i \in A} \sum_{j \in B} w_{i,j}$ and $d_A = \sum_{i \in A} d_i$. *Ratio Cut* addresses the balance concern on the sizes of partitioned graph ([68]), which leads to minimization of the following objective function

$$\mathcal{J}_{RCut} = \frac{S(A,B)}{|A|} + \frac{S(A,B)}{|B|}$$

Normalized Cut addresses the balance concern on the weights of partitioned graph ([134]), which leads to minimization of

$$\mathcal{J}_{NCut} = \frac{S(A,B)}{d_A} + \frac{S(A,B)}{d_B}$$

Min-Max Cut addresses the balance concern between the intra-cluster weights and inter-cluster weights in a partitioning ([49]), which leads to minimization of

$$\mathcal{J}_{MCut} = \frac{S(A,B)}{S(A,A)} + \frac{S(A,B)}{S(B,B)}$$

By relaxing cluster memberships to real values, the above minimization problems

can all be formulated as eigenvector problems related to the graph Laplacian

$$\begin{aligned} \mathcal{J}_{RCut} &= \mathbf{q}^\top \mathbf{L}\mathbf{q} \\ \mathcal{J}_{NCut} &= \mathbf{q}^\top \tilde{\mathbf{L}}\mathbf{q} \\ \mathcal{J}_{MCut} &= \frac{\mathbf{q}^\top \mathbf{W}\mathbf{q}}{\mathbf{q}^\top \mathbf{D}\mathbf{q}} \end{aligned}$$

where $\mathbf{q}$ is related to the relaxed cluster membership. All the above three problems lead to finding the second eigenvector of the graph Laplacian $\mathbf{L}$ or $\tilde{\mathbf{L}}$.

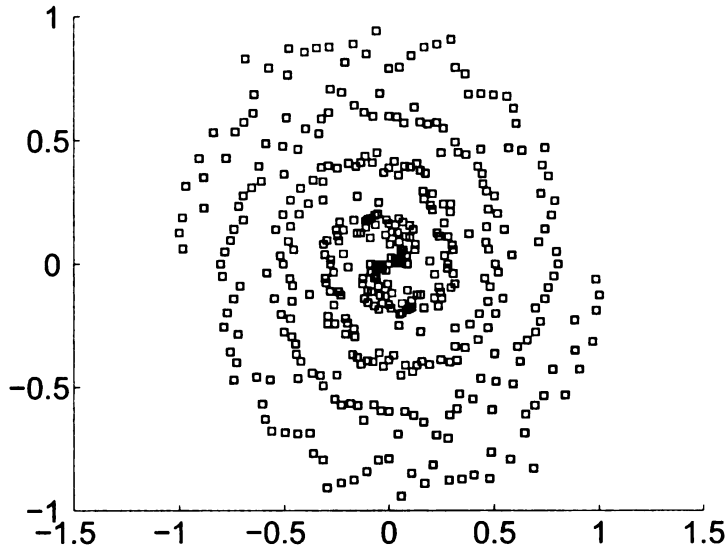
2-way spectral clustering can be extended to k-way spectral clustering ([66, 115]), whose solution is related to the top k eigenvectors of the graph Laplacian.

1.2 Side Information

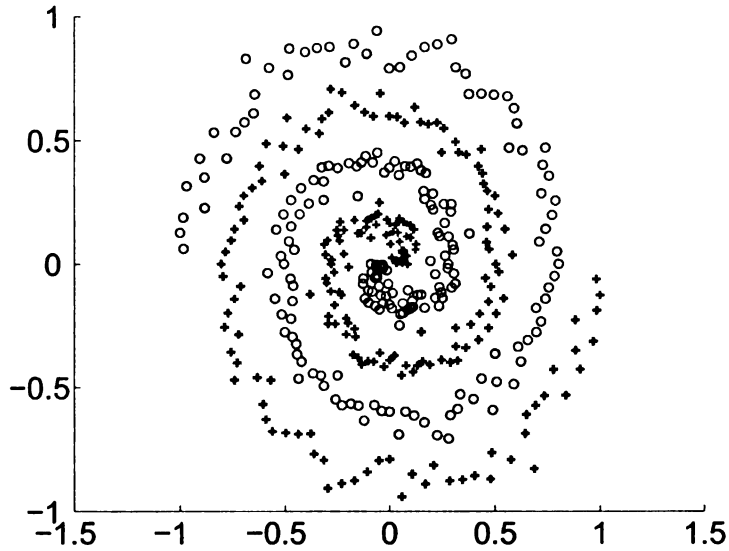
1.2.1 Motivation on Introducing Side Information

In the previous section, we briefly reviewed graph-based semi-supervised learning. All the models mentioned there only assume the knowledge of unlabeled data, or more specifically, the input patterns of unlabeled data. As we can see, by constructing a graph from all the data, those models do utilize unlabeled data in training. However on many occasions, it is hard to achieve satisfying performance, even with the knowledge of input patterns of both labeled and unlabeled data. The difficulties come from various sources. To name a few here

- Very small number of training examples. As a result, not enough supervised information can be gained.
- Large number of classes with skewed size distributions. For example, it is easy to come across hundreds of categories in text classifications problems. Very often, in those categories only a few major ones are of large sizes, while the rest are minor categories with relatively smaller sizes. Classifiers tend to make mistakes on those minor categories.

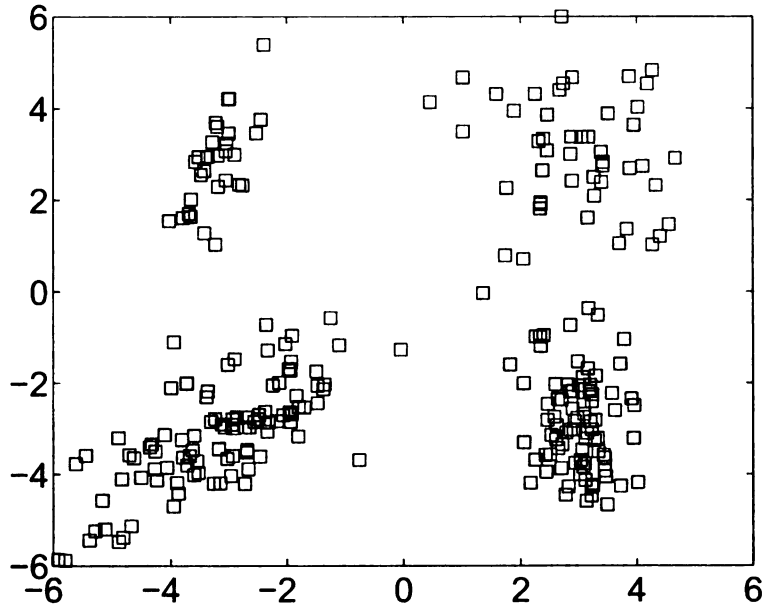


(a) Dataset 1

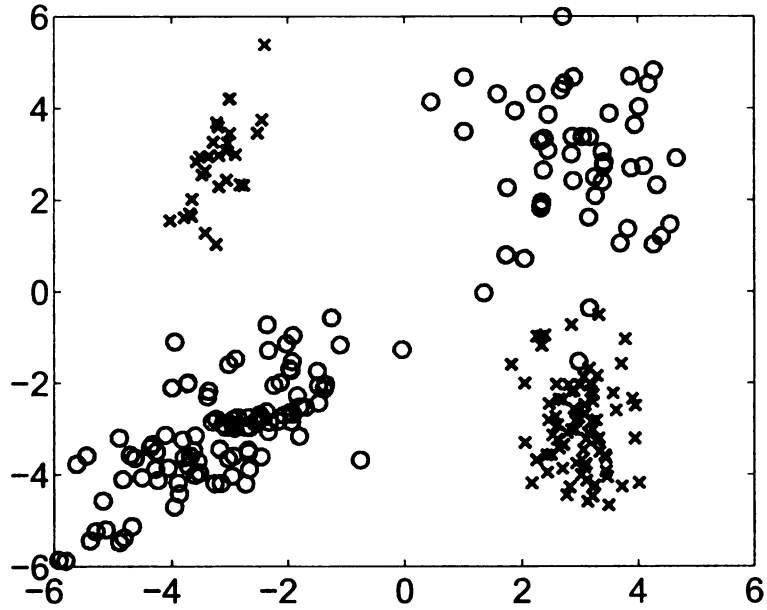


(b) Dataset 1 (with true class information)

Figure 1.1. Example dataset that presents data in complicated structure. The upper figure shows the data distributions without class information; the lower figure shows class lables for data examples. Without any supervised information, clustering on this dataset is difficult, since the two spirals are hard to be separated.



(a) Dataset 2



(b) Dataset 2 (with true class information)

Figure 1.2. Example dataset that presents data in complicated structure. The upper figure shows the data distributions without class information; the lower figure shows class lables for data examples. Without any supervised information, 2-way clustering on this dataset is difficult, since it is hard to tell how to combine four data clouds into two clusters.

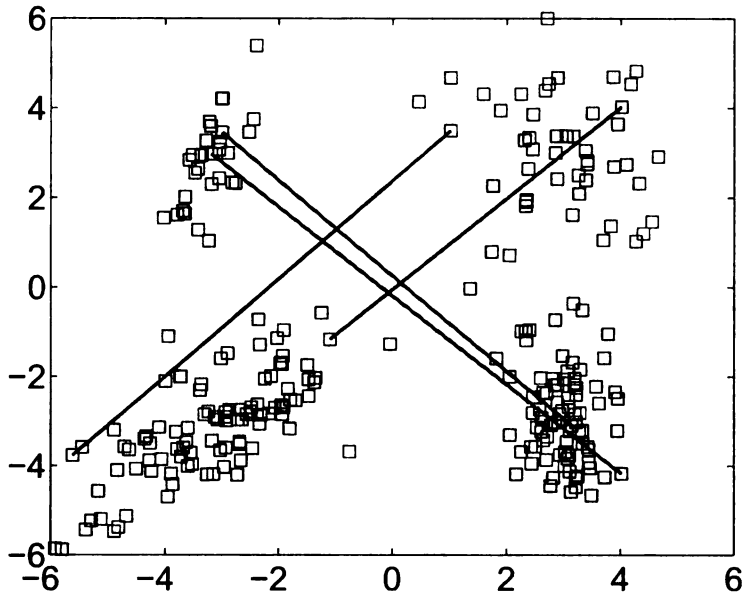


Figure 1.3. Dataset 2 with side information.

- Complicated underlying structure of data. For example, in general a clustering algorithm is hard to perform well when the data examples do not form compact and well-separated clusters. Figure 1.1 and Figure 1.2 present two such cases that will fail many clustering algorithms. It is easy to imagine, yet hard to visualize, the existence of much more complicated structure of data in real-world applications.

- Noisy input patterns. As a result, the correlations between data examples computed from their input patterns are unreliable, so propagating information among data examples could be error-prone.

- Incomplete data. In this situation, it is hard to reveal the entire structure of data.

However, with a little more information known beyond the data input patterns, the difficulty involved in the semi-supervised learning can be eased or resolved. To

illustrate this finding, let us consider the 2-way clustering problem shown in Figure 1.2. It is easy to find that there are roughly four clouds of data points in the dataset, but the difficult part is in what way these four clouds should be combined into two clusters. Suppose we know that there a few data pairs, each of them containing two data points that should be put into the same cluster, as indicated by solid lines in Figure 1.3. Then it is clear that the two data clusters should be formed by putting the two clouds in the diagonal positions into one cluster.

In the above example, the extra information is only data relationships that take only binary values, from merely four data pairs involving eight data points. Considering the input patterns of all the data points and the relationships between all possible data pairs, the extra information we additionally have is very simple and small, so we can call it as a type of "side information". However, the data clustering task is greatly benefited from such side information.

1.2.2 Side Information for Semi-supervised Learning

We may find different forms of side information in different task scenarios. Numerous previous studies have resorted to side information, in order to improve the performance of learning algorithms. For example

- The title, text annotations, or even surrounding texts are often referred as side information to the image representations. As is well known, understanding image contents is very hard due to the gap between the low-level image features and the high-level image semantics. The textual side information will provide additional features that help to describe the image contents. Precious work along this line includes using textual information for image segmentation [10], image retrieval [11, 35, 170], and image clustering [31].

- Log data is widely gathered in web applications (such as search engine, information filtering, online shopping, and etc.) and serves as side information for

various purposes. User query log can be used to form a context to provide additional information on the user's information need, and then improve the query model ([139, 131]) or the user model ([132]). User click-through data [1] is another important source of side information for web applications. For example, from click-through data we can tell how often a link from page A to page B is actually used, which can be used to estimate the similarity between the two pages [112], in addition to their page contents; gathering a specific user's clicked links can help to model the user's interest [75, 140], as additional information to the user's profile.

- User feedback can be regarded as side information for information retrieval. For example, user ratings on a returned document help to decide its relevance to the query. In movie recommendation websites, user ratings provide additional information either for describing movies or describing users.

- External linguistic resource can serve as side information for many information retrieval applications. For example, bilingual dictionaries are widely used to help query translation for cross-language information retrieval [2, 95, 46, 57, 76, 93, 109, 81]. WordNet [54] can be used for query expansion, word disambiguation, etc., and finally improve retrieval performance [106, 147].

- For classification problems, knowledge gained about the categories can be viewed as side information. For example, human knowledge on the popularity of each category is always used to gain some control on the category sizes in the classification results. In image databases such as ImageCLEF [77], there is some text description associated with each image category, which can be used to gain some knowledge on the relationships between categories.

[1] Click-through data refers to a type of log which keeps record of the time, session of hyperlinks being clicked by web users.

- In real-world data clustering problems, it is often possible to gather information on whether two examples should be put in the same cluster or not (often called as "pairwise constraints"). For example, in speaker segmentation and recognition, a conversation between several speakers needs to be segmented and clustered according to the speaker identity. It may be possible to automatically identify small segments of speech which are likely to be from a single unknown speakers [9]. Another example would be finding lanes in GPS maps. Usually lanes are identified by clustering a few road segments. By tracking the signals of car GPS's, continuous car-traveled segments would be considered belonging to the same lane, and well separated car-traveled segments would be considered belonging to different lanes [148]. These pairwise constraints, would provide side information for clustering data examples based on their input patterns.

- In some cases, a few identified representative data examples are available as "seeds" for clustering. We can also view these cluster seeds as side information [13, 12].

One characteristic can be summarized from various forms of side information is that, it is capable of providing some additional knowledge about the data, but it cannot replace the original data input patterns. In this thesis, we will refer to *any additional information on the unlabeled data beyond input patterns* as "side information".

Usually side information is small in amount and incomplete. However, with proper use, side information could significantly improve the performance of unsupervised, supervised, or semi-supervised learning algorithms.

Side information serves as partial supervised information. Also, as mentioned in the beginning of this chapter, we will assume the data input patterns of all the data (including labeled and unlabeled) is known; and in all following discussions, we will use unlabeled data in training. For these reasons, regardless of the original learning

task being an unsupervised one or a supervised one, as long as side information is used, we will refer the resulting learning method as **semi-supervised learning with side information**.

Side information can be explored in a number of different ways. For instance, in probabilistic models, side information is always represented as strong priors; in non-probabilistic models, side information is always incorporated in distance metrics or constraints, etc. The work presented in the thesis will focus on graph-based non-probabilistic models.

1.3 Graph-based Semi-supervised Learning with Side Information

Side information, depending on its form, has various interpretations in graph-based semi-supervised learning models. These interpretations can be summarized into, but not limited to, the following major categories

1. **Additional dimensions to the node representation:** more features are found for the input pattern of part (or all) the data examples. A typical example in this category is textual information for images, where we can append textual feature vectors with image feature vectors to form a new image representation. Other examples include query log for query modeling, click-through data for user modeling, user ratings for movie recommendation, etc.

2. **Constraints on the graph topology:** an edge should or should not be constructed between two nodes. The pairwise constraints for clustering belongs to this category, which is self-explanatory. For example, the link graph constructed for web pages by hyperlinks falls into this category.

3. **Refinement on graph edge weights:** a (new) weight of an edge is proposed. For instance, user feedback on a returned document defines a new relevance

score between the query and the document, i.e. a similarity value between the query node and the document node. Click-through data for hyperlinks on a web page also suggests a new weight for the corresponding pair of page nodes (for example, proportional to the number of user clicks on the hyperlinks).

The variance in the roles played by side information from a graph point of view leads to a diversity of ways to accommodate side information within semi-supervised learning settings. The following is an incomplete list of treatments of side information

1. **Data Representation Augmentation:** Side information is used to form better representation, in the hope that the underlying structure of data can be better revealed when the augmented data representation is used for computation. For example, in [63, 32, 120, 130, 33, 118], side information is used to generate additional feature dimensions for classification; in [48, 40], side information is used to learn a new representation in the latent space.

2. **Hard Constraints:** Side information is formulated into constraints that cannot be violated. For example, constrained clustering [148, 20], constrained EM algorithm [73] and multiple-instance learning [6] all treat side information as non-violable rules.

3. **Soft Penalty:** Any inconsistency between the learning outcome and the side information will incur a penalty. For those models with a clear optimization goal, such a treatment often leads to side information being encoded as part of the optimization objective function (such as a regularizer term or its equivalence). Typical examples include using side information for distance metric learning [155, 98, 74], clustering [96, 15, 108, 16], etc..

4. **Other Heuristics to Carry Out Learning Algorithms:** Side information is used to provide building blocks for other learning algorithms. For example,

in [168, 167, 119, 70, 158, 149], side information is used to induce the graph structure where learning algorithms over the graph can be carried out; in [27, 39] side information is used for co-training; in [72] side information is used for boosting.

1.4 Overview on Thesis Work

All the thesis work presented in the rest of the chapters revolves around the topic of improving semi-supervised learning with side information. Due to the diversity of learning tasks and forms of side information, we will demonstrate our efforts through several typical learning settings and applications.

Chapter 2 discusses multi-label classification, with class correlations as side information. To meet the challenge of a large number of classes and a small size of training data in multi-label classification, we propose to utilize class correlations to link the computation of data example similarities with their multiple class memberships. We will also show that the proposed multi-label learning algorithm leads to a constrained non-negative matrix factorization formalization.

Chapter 3 focuses on binary classification, with side information in the form of links. Link-based classification gained significant attention in text domains, due to the large amount of needs in classifying web pages or scientific publications, where inter-document connections are available as hyperlinks or references. However, the sparse and noisy nature of links causes trouble in utilizing them for classification. A general boosting framework is proposed in this chapter that tries to make the best use of link information against its sparseness and noise, and is able to improve *any* binary classifier.

Chapter 4 presents another boosting framework for semi-supervised clustering, with pairwise constraints as side information. In this work, side information is encoded into the data representations by iteratively selecting a good direction to project

the original data into a low-dimensional space. Again, the proposed framework is designed as a meta-algorithm that can be applicable to *any* clustering algorithm, and improve its performance with pairwise constraints.

In addition to the above generic tasks of semi-supervised learning, the thesis work also includes two application examples. In particular, we will show two information retrieval tasks, where semi-supervised learning with side information could achieve improved performance over traditional methods. In Chapter 5, we study the problem of cross-language information retrieval, with bilingual dictionaries as side information. Two models based on a maximum coherence principle are proposed, which can be well explained as two-way partitioning of the graph induced by side information – the dictionary.

Another application, extracting question-answer pairs from web FAQs, is described in Chapter 6, where side information comes from human knowledge on the presentation regularities of web FAQs. The model proposed in this chapter leads to a correlated label propagation scheme over a graph built upon the text segments of web FAQ pages. As will be seen, properly using the side information enables the question-answer extraction task being performed without any human supervision, despite the wide variety of possible contents and styles in web pages.

Finally, Chapter 7 concludes the thesis work.

Semi-supervised learning enforces the predicted labels (usually the learning goal) to be consistent with the structure of data, while agreeing with the supervised information (if available). This is already stated as the assumption made by all graph-based learning algorithms in Section 1.1.2. The introduction of side information should not violate this data consistency assumption. Therefore, including side information into the consistency enforcement becomes necessary. Such an idea is common across all the chapters that follow, in despite of the diversity of tasks, models and their graph interpretations that will be discussed in detail in each individual chapter.

In summary, the semi-supervised learning models and frameworks proposed in this thesis work all conform to the same theme: *consistency is maximally enforced among the following factors – data input patterns, supervised information (if any), side information, and predictions on unlabeled data*; invariably, optimization will be the tool to achieve the goal of consistency enforcement. Before going into detailed discussions in each chapter, understanding this theme will help to reveal the connections between all parts of the thesis work and view them as one whole piece.

CHAPTER 2

Semi-supervised Multi-label Learning with Class Correlations

2.1 Introduction

Multi-label learning refers to the classification problems where each example can be assigned to multiple different classes. It has found applications in many real-world problems. For example, text categorization is typically multi-labeled since each document can be assigned to several predefined topics; in bioinformatics, most genes are associated with more than one functional classes (e.g., metabolism, transcription and protein synthesis); automatic image annotation, can also be treated as a multi-label learning problem if we view each annotation word as a distinct class.

A straightforward approach toward multi-label learning is to decompose it into a set of binary classification problems, one for each class. The drawback with this approach is that it does not explore the correlation among different classes, which often could be an important hint for deciding the class memberships. Many algorithms have been developed to incorporate the class correlation into multi-label learning, including [110, 52, 85, 146, 28, 59, 60, 90, 171, 164, 145, 141, 42]. But most of these studies are limited to a relatively small number of classes and assume that the amount of training data is sufficient for exploiting class correlations and training

reliable classifiers. In contrast, the real-world application of multi-label learning often features a large number of classes and a relatively small size of training data. As a result, the amount of training data related to each class is often sparse and insufficient for computing class correlations and learning a reliable classifier.

However, we find that in practical, more reliable class correlations are often available as side information. For example, in text classification problems, if each category has a description, the correlation between two classes can be derived by computing the similarity between their descrptions; also, human knowledge about the category topic can also be used to decide how close two categories are in terms of their topics. So the problem becomes, *given class correlation information*, how can we improve multi-label learning?

To address this problem, we present a novel framework for multi-label learning that explicitly explores the correlation among different classes. Compared to the existing approaches for multi-label learning that also explore the class correlation, the proposed framework provides a natural means for exploring the unlabeled data and the class correlation simultaneously, thus effective for the learning scenarios with a large number of classes and a small size of training data.

The key assumption behind this work is that two examples tend to have large overlap in their assigned class memberships if they share high similarity in their input patterns. To be more specific, consider two examples $\mathbf{x}_1$ and $\mathbf{x}_2$ that are labeled by two sets of class labels $\mathbf{y}_1$ and $\mathbf{y}_2$, respectively. We can evaluate the similarity between these two examples in two different ways. The first one is based on the correlation between the input patterns of these two examples. The second one is based on the overlap between the class labels of these two examples. We denote the similarity based on the input patterns by $K_x(\mathbf{x}_1, \mathbf{x}_2)$, and the similarity based on the class labels by $K_y(\mathbf{y}_1, \mathbf{y}_2)$. If the assigned class labels $\mathbf{y}_1$ and $\mathbf{y}_2$ are appropriate for example $\mathbf{x}_1$ and $\mathbf{x}_2$, we would expect the two similarity measurements to be similar,

namely $K_x(\mathbf{x}_1, \mathbf{x}_2) \approx K_y(\mathbf{y}_1, \mathbf{y}_2)$. Based on this assumption, we can determine the best assignment of class labels to the unlabeled data by minimizing the difference between the two sets of similarities. Clearly, this approach is able to effectively explore the unlabeled data because the assignment of class labels to each unlabeled example is dependent on the assignment of class labels of other unlabeled examples. This approach is also able to exploit the class correlation effectively through the kernel similarity function $K_y(\mathbf{y}_1, \mathbf{y}_2)$.

The rest of the chapter is structured as follows: first, we briefly review the related work on multi-label learning and semi-supervised learning; second, we introduce the proposed framework for multi-label learning, and a formalization based on the constrained non-negative matrix factorization; third, we present an efficient algorithm to solve the related optimization problem that is based on the iterative bound optimization algorithm; fourth, we present the empirical study with a text categorization problem; finally, we conclude this study and raise some future work.

2.2 Related Work

We will first review the related work on multi-label learning, followed by a discussion of related semi-supervised learning problem.

2.2.1 Related Work in Multi-label Learning

The simplest approach toward multi-label learning is to divide it into a number of binary classification problems [160, 86]. There are a number of disadvantages with this approach. One is that it will not scale to a large number of classes since a different binary classifier has to be built for each class. Another disadvantage is that it treats each class independently, and therefore is unable to explore the correlation among different classes. The third disadvantage is that this approach often will suffer from the unbalanced data problem when the minority classes are given only a few training

examples.

Another group of approaches toward multi-label learning is label ranking [42, 52, 127]. Instead of learning binary classifiers from labeled examples, these approaches learn a ranking function from the labeled examples that order class labels for a given test example according to their relevance to the example. Compared to the binary classification approaches, the label ranking approaches are advantageous in dealing with large numbers of classes because only a single ranking function is learned. However, similar to the binary classification approaches, the label ranking approaches are usually unable to exploit the class correlation information.

In the past, a number of studies have been devoted to exploring the class correlation within the context of multi-label learning. A generative model for multi-label learning was proposed in [146] to explicitly incorporate the pairwise correlation between any two class labels. A maximum entropy model is proposed in [171] that capture the pairwise class correlation by constraints. Approaches based on latent variables were proposed in [110, 164] to capture the correlation among different classes. The study in [123] exploited the class correlation information given the hierarchical structure of classes. Unlike the previous work on multi-label learning that only considers the correlation among different classes, in this chapter, we present a novel framework that exploits the unlabeled data as well as the class correlation. This property will make the proposed approach more effective than the existing approaches for multi-label learning, particularly when the number of classes is large and the size of training data is small.

This work is also particularly related to the label propagation approaches for semi-supervised learning. This is because by enforcing examples with similar input patterns to share similar sets of class labels, we essentially propagate the class labels through the similarity graph of examples, which is the key idea of the label propagation approaches. A number of machine learning methods have been developed recently

for label propagation, including the Gaussian processes [152], the harmonic functions [173], and Green functions [168]. Unlike most of the previous work on semi-supervised learning that is designed primarily for multi-class learning, this work is specifically targeted on the semi-supervised multi-label learning. It effectively explores the class correlation information when utilizing the unlabeled data. More discussion of semi-supervised learning can be found in [129, 172].

2.2.2 Representative Algorithms

In the following, we will recapitulate a few representative algorithms for semi-supervised multi-label learning, as well as the non-negative matrix factorization algorithm that is closely related to the model we are going to propose later.

Multi-class Multi-label Perceptron Algorithm

Multi-class Multi-label Perceptron (MMP) algorithm extends the Perceptron algorithm from single binary output to a ranked list of n-ary output. Specifically, for any test example possibly belonging to one or more of l categories, the algorithm output will be a ranked list of the l categories, indicating the preference of assigning the test example to those categories.

As the Perceptron algorithm, MMP maintains a set of l prototypes $\mathbf{w}_1, \cdots, \mathbf{w}_l$. For any data example $\mathbf{x}_j$, $\mathbf{w_i}^\top \mathbf{x}_j$ yields a score that will decide the ranking of the i-th category for this data example x_j. The procedures of learning those prototypes can be summarized as follows

Initialize: Set $\mathbf{w}_1 = \cdots = \mathbf{w}_l = \mathbf{0}$

Loop:

- Get a new training data example $\mathbf{x}_j \in \mathbb{R}^d$, and its true category information $\hat{\mathbf{y}}_j$, which is a set of category IDs

- Rank the categories according to $\mathbf{w_1}^\top \mathbf{x}_j, \cdots, \mathbf{w_i}^\top \mathbf{x}_l$
- If the category ranking is not consistent with the truth $\hat{\mathbf{y}}_j$
  1. For $\forall r \in \hat{\mathbf{y}}_j$, set $n_r = |\{s \notin \hat{\mathbf{y}}_j | \mathbf{w}_s^\top \mathbf{x}_j \geq \mathbf{w}_r^\top \mathbf{x}_j\}|$
  2. For $\forall r \notin \hat{\mathbf{y}}_j$, set $n_r = |\{s \in \hat{\mathbf{y}}_j | \mathbf{w}_s^\top \mathbf{x}_j \leq \mathbf{w}_r^\top \mathbf{x}_j\}|$
  3. Compute loss $\eta = \sum_r n_r$
  4. Update for $r \in \hat{\mathbf{y}}_j$: $\mathbf{w}_r \leftarrow \mathbf{w}_r + \frac{n_r}{\eta}\mathbf{x}_j$
  5. Update for $r \notin \hat{\mathbf{y}}_j$: $\mathbf{w}_r \leftarrow \mathbf{w}_r - \frac{n_r}{\eta}\mathbf{x}_j$

Output: $\mathbf{w}_1, \cdots, \mathbf{w}_l$

MMP algorithm is computationally inexpensive and it is suitable for online learning.

Multi-label Informed Latent Semantic Indexing Algorithm

Multi-label Informed Latent Semantic Indexing (MLSI) algorithm extends the Latent Semantic Indexing (LSI) to supervised cases [164]. In LSI, a linear mapping from the input space to a low-dimensional latent space is found, so that the structure of the data can be preserved as much as possible. This leads to an optimization problem which minimizes the reconstruction error from the latent space to the original space. Given supervised information in the form of multiple labels of training examples, MLSI proposes to preserve the structures in the input patterns of the data, as well as their label information, using the same set of latent variables.

Formally, let $\mathbf{X} \in \mathbb{R}^{n \times d}$ denote the input matrix for n data examples, each in d dimensions; let $\mathbf{Y} \in \mathbb{R}^{n \times l}$ denote the corresponding label indicator matrix. The MLSI algorithm is equivalent to solve the following optimization problem

$$\begin{aligned} \min_{\mathbf{A},\mathbf{B},\mathbf{V}} \quad & (1-\beta)||\mathbf{X} - \mathbf{V}\mathbf{A}||^2 + \beta||\mathbf{Y} - \mathbf{V}\mathbf{B}||^2 \\ s.t. \quad & \mathbf{V}^\top \mathbf{V} = \mathbf{I} \end{aligned}$$

where $\mathbf{V} \in \mathbb{R}^{n \times k}$ defines a latent space, which gives a k-dimensional projection for both the input pattern (i.e. $\mathbf{X}$) and their labels (i.e. $\mathbf{Y}$); $\mathbf{A} \in \mathbb{R}^{k \times d}$ and $\mathbf{B} \in \mathbb{R}^{k \times l}$ give the mappings from the latent space to the input space and the output space (i.e. labels), respectively; β is a constant which balances the reconstruction errors to both the input spaces and output spaces. The above optimization problem is equivalent to the following eigenvector problem

$$\begin{aligned} \max_{\mathbf{v} \in \mathbb{R}^n} \quad & \mathbf{v}^\top \mathbf{C} \mathbf{v} \\ s.t. \quad & \mathbf{v}^\top \mathbf{v} = 1 \end{aligned}$$

where $\mathbf{C} = (1 - \beta)\mathbf{X}\mathbf{X}^\top + \beta \mathbf{Y}\mathbf{Y}^\top$. The solution of the latent semantic matrix $\mathbf{V}$ is composed of the first k-th eigenvector of the above eigenvector problem, i.e., $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_k]$.

Note that, MLSI includes output variable $\mathbf{Y}$ in its objective function. The corresponding term can be rewritten as

$$\begin{aligned} \beta ||\mathbf{Y} - \mathbf{V}\mathbf{B}||^2 &= \beta \cdot trace[(\mathbf{Y} - \mathbf{V}\mathbf{B})^\top (\mathbf{Y} - \mathbf{V}\mathbf{B})] \\ &= \beta \cdot trace[\mathbf{Y}^\top \mathbf{Y} - \mathbf{B}^\top \mathbf{V}^\top \mathbf{Y} - \mathbf{Y}^\top \mathbf{V}\mathbf{B} + (\mathbf{V}\mathbf{B})^\top \mathbf{V}\mathbf{B}] \end{aligned}$$

In the above, $\mathbf{Y}^\top \mathbf{Y}$ can be interpretted as a class similarity matrix, computed solely from the training data. Therefore, MLSI implicitly addresses the idea of utilizing class correlation. However, since only training data is involved, when the training data is small in amount, such class similarity information could not be very reliable.

Non-nongative Matrix Factorization

When we use a matrix to represent a graph, such as the adjacency matrix or other variants, there is a potential computation burden when the graph has a large number nodes and edges. A natural thought would be to approximate the matrix while preserving as much information as possible. This is particularly useful especially when we are more interested in understanding global structures of the data.

Finding the optimal k-rank approximation of a given r-rank matrix $\mathbf{A}$ ($k < r$) can be formulated as ([138, 163])

$$\mathbf{B} = \underset{Rank(\mathbf{B})=k}{\arg\min} ||\mathbf{A} - \mathbf{B}||_F \tag{2.1}$$

Applying Singular Value Decomposition to matrix $\mathbf{A}$, we have

$$\mathbf{A} = \mathbf{USV}^\top$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices, and $\mathbf{S} = diag(s_1, s_2, \ldots, s_r, 0, \ldots, 0)$ with $s_1 \geq s_2 \geq \cdots \geq s_r > 0$, the solution to the lower rank approximation problem would be

$$\mathbf{B} = \mathbf{U}_k diag(s_1, s_2, \ldots, s_k)\mathbf{V}_k^\top \tag{2.2}$$

Highly related to matrix approximation problem, non-negative matrix factorization tries to approximate a matrix $\mathbf{A}$ with two non-negative matrix factors $\mathbf{U}$ and $\mathbf{V}$ ([104, 105])

$$\mathbf{A} \approx \mathbf{UV}$$

To measure the approximation quality, two cost functions are used. The first measurement is the square of Euclidean distance

$$||\mathbf{A} - \mathbf{B}|| = \sum_{i,j}(A_{i,j} - B_{i,j})^2$$

and the second measurement is the divergence

$$D(\mathbf{A}||\mathbf{B}) = \sum_{i,j}\left(A_{i,j}\log\frac{A_{i,j}}{B_{i,j}} - A_{i,j} + B_{i,j}\right)$$

Note that the second measurement $D(\mathbf{A}||\mathbf{B})$ is always nonnegative and reaches zero only when $A_{i,j} = B_{i,j}$ holds for all (i, j) pairs.

An iterative algorithm has been proposed to efficiently solve the problem by iteratively minimizing the above two cost functions. In particular, the following updating rule minimizes the Euclidean distance $||\mathbf{A} - \mathbf{UV}||$

$$U_{i,a} \leftarrow U_{i,a} \sum_k \frac{A_{i,k}}{(\mathbf{UV})_{i,k}} V_{a,k}$$
$$U_{i,a} \leftarrow \frac{U_{i,a}}{\sum_j U_{j,a}}$$

and the following updating rule minimizes the divergence $D(\mathbf{A}||\mathbf{UV})$

$$V_{a,k} \leftarrow V_{a,k} \sum_i U_{i,a} \frac{A_{i,k}}{(\mathbf{UV})_{i,k}}$$

The above two rules which guarantees the two cost functions to be non-increasing until a local optimum is reached. To initialize the algorithm, the two matrix factors $\mathbf{U}$ and $\mathbf{V}$ can be seeded as random non-negative matrices.

2.3 Semi-Supervised Multi-label Learning by Constrained NMF

The following terminology and notations will be used throughout the rest of the chapter. Let $\mathcal{D} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ denote the entire dataset, where n is the total number of examples, including both the labeled ones and the unlabeled ones. We assume that the first n_l examples are labeled ones, and their label information is presented in the binary matrix $\bar{\mathbf{T}} \in \{0,1\}^{n_l \times m}$ where m is the number of classes. Let the similarity of all the examples denoted by a matrix $\mathbf{A} = [A_{i,j}]_{n \times n}$, where element $A_{i,j} \geq 0$ represents the similarity between two examples based on their input patterns. We denote by $T_{i,k} \geq 0$ the confidence score of assigning the k-th class label to the i-th example, and by $\mathbf{t}_i = (T_{i,1}, T_{i,2}, \ldots, T_{i,m})^\top$ the confidence scores of assigning each class to the i-th example. Finally, the matrix $\mathbf{T} = [T_{i,k}]_{n \times m}$ denotes the confidence scores of assigning every class label to all examples.

2.3.1 A Framework for Multi-label Learning

The key assumption behind this work is that two examples tend to be assigned similar sets of class labels if they share high similarity in the input patterns. In order to utilize this assumption for predicting class labels, we evaluate the similarity of two examples in different ways, one by their input patterns and the other by their assigned class memberships. We refer to the former as the *input-based similarity*, and the latter as the *class-based similarity*. Then, if the class labels assigned to the examples are consistent with their input patterns, we would expect the class-based similarities to be close to the input-based similarities. Since the input-based similarities are already given by the matrix $\mathbf{A}$, the key question is how to compute the similarity of two examples based on their class memberships. The simplest approach is to compute the class-based similarity between examples $\mathbf{x}_i$ and $\mathbf{x}_j$ by the overlap between their classmemberships, or $\mathbf{t}_i^\top \mathbf{t}_j$. The problem with this similarity measurement is that it treats all the classes independently and therefore is unable to explore the correlation among them. In particular, it will give zero similarity whenever two examples share no common classes. However, two examples with no common shared classes can still be strongly related if their assigned classes have close relationship (e.g. the children-parent relationship in the hierarchy of class labels).

To capture the side information of correlation among different classes, we introduce matrix $\mathbf{B} = [B_{k,l}]_{m \times m}$ for the class similarities. Each element $B_{k,l} \geq 0$ represents the similarity between two classes. Then, instead of computing the class-based similarity between two examples by the direct dot product, we compute it by a weighted dot product, i.e., $\mathbf{t}_i^\top \mathbf{B} \mathbf{t}_j$. Then, following the assumption stated above, we would expect $A_{i,j} \approx \mathbf{t}_i^\top \mathbf{B} \mathbf{t}_j$ if the class assignments $\mathbf{t}_i$ and $\mathbf{t}_j$ are appropriate for examples

$\mathbf{x}_i$ and $\mathbf{x}_j$. This leads to the following optimization problem:

$$\begin{aligned} \arg\min_{\mathbf{T}} \quad & \sum_{i,j=1}^{n} \left(A_{i,j} - \sum_{k,l=1}^{m} T_{i,k} B_{k,l} T_{j,l} \right)^2 && (2.3) \\ \text{s. t.} \quad & T_{j,l} \geq 0,\ j = 1, \ldots, n,\ l = 1, \ldots, m \\ & T_{i,k} = \bar{T}_{i,k},\ i = 1, \ldots, n_l,\ k = 1, \ldots, m && (2.4) \end{aligned}$$

where the last set of constraints is to ensure that the estimated label confidences $T_{i,k}$'s are consistent with the assigned class labels $\bar{T}_{i,k}$'s for all the training examples.

Remark: It is interesting to see that the formalization in (2.3) can also be written as a non-negative matrix factorization problem under a linear constraint, if we ignore the constraints coming from the training examples, i.e.,

$$\begin{aligned} \arg\min_{\mathbf{T},\mathbf{H}} \quad & \|\mathbf{A} - \mathbf{T}\mathbf{H}\|_F \\ \text{s. t.} \quad & T_{j,l}, H_{j,l} \geq 0,\ j = 1, \ldots, n,\ l = 1, \ldots, m \\ & \mathbf{H} = \mathbf{B}\mathbf{T}^\top \end{aligned}$$

where $\|\cdot\|_F$ stands for the Frobenius norm. The above problem is similar to the standard Non-negative Matrix Factorization (NMF) problem except for the linear constraint that restricts the matrix $\mathbf{H}$ to be linearly dependent on the matrix $\mathbf{T}$. It is this constraint and furthermore the constraints arising from the labeled data that prevent the direct application of the NMF algorithm.

One problem with the formulation in (2.3) is that since the input-based similarity $A_{i,k}$ can be any positive value, it could be significantly larger than the elements in $\mathbf{B}$. As a result, the label confidence $T_{i,k}$ that minimizes the objective function in (2.3) will also be significantly larger than 1. But, to satisfy the constraints in (2.4), the label confidence $T_{i,k}$ should be restricted to 0 or 1 since the assigned class label $\bar{T}_{i,k}$ is binary. To resolve the conflicts between the minimizer of the objective function in (2.3) and the binary constraints, we introduce two sets of label confidences: the unnormalized label confidence $\{T_{i,k}\}$, and the normalized label confidence $\{\hat{T}_{i,k}\}$.

The former can take any positive value, while the latter is positive and subject to the constraints of $\sum_{k=1}^{m} \hat{T}_{i,k} = 1$. We will on one hand, use the unnormalized label confidence to minimize the difference between the class-based similarity and the input-based similarity, and on the other hand, use the normalized label confidence to ensure that the predicted label confidence is consistent with the assigned class labels. Formally, we can summarize this idea into the following optimization problem:

$$\begin{aligned} \underset{\mathbf{T},\hat{\mathbf{T}},\alpha}{\arg\min} \quad & \sum_{i,j=1}^{n} \left(A_{i,j} - \sum_{k,l=1}^{m} T_{i,k} B_{k,l} T_{j,l} \right)^2 + C \sum_{j=1}^{n} \sum_{l=1}^{m} (T_{j,l} - \alpha_j \hat{T}_{j,l})^2 \quad (2.5) \\ \text{s. t.} \quad & T_{j,l}, \hat{T}_{j,l}, \alpha_j \geq 0,\ j = 1, \ldots, n,\ l = 1, \ldots, m \\ & \sum_{l=1}^{m} \hat{T}_{j,l} = 1,\ i = 1, \ldots, m \\ & \hat{T}_{i,k} = \frac{\bar{T}_{i,k}}{\sum_{k=1}^{m} \bar{T}_{i,k}},\ i = 1, \ldots, n_l,\ k = 1, \ldots, m \end{aligned}$$

Note that in the above formalization, we introduce the term $C \sum_{j=1}^{n} \sum_{l=1}^{m} (T_{j,l} - \alpha_j \hat{T}_{j,l})^2$ into the objective function to enforce that the two sets of label confidences are consistent and only differ by a scaling factor α_j for each example. Parameter C weights the importance of the second term against the first term, and is determined empirically.

2.4 Solving the Constrained NMF

An alternative optimization approach is adopted to solve the constrained NMF. In particular, we will solve the optimization problem by alternatively fixing one set of label confidences and finding the optimal solution for another set of label confidences.

More specifically, we first fix the normalized label confidence matrix $\hat{\mathbf{T}}$ and the scaling factors α_j's, and search for the unnormalized label confidence $T_{i,j}$ that optimizes (2.5). To this end, we upper-bound the term $\left(A_{i,j} - \sum_{k,l=1}^{m} T_{i,k} B_{k,l} T_{j,l} \right)^2$

as follows

$$
\begin{aligned}
&\left(A_{i,j} - \sum_{k,l=1}^{m} T_{i,k}B_{k,l}T_{j,l}\right)^2 \\
\leq\ & \sum_{k,l=1}^{m} \frac{\tilde{T}_{i,k}B_{k,l}\tilde{T}_{j,l}}{[\tilde{\mathbf{T}}\mathbf{B}\tilde{\mathbf{T}}^\top]_{i,j}} \left(A_{i,j} - [\tilde{\mathbf{T}}\mathbf{B}\tilde{\mathbf{T}}^\top]_{i,j}\frac{T_{i,k}B_{k,l}T_{j,l}}{\tilde{T}_{i,k}B_{k,l}\tilde{T}_{j,l}}\right)^2 \\
=\ & A_{i,j}^2 + \sum_{k,l=1}^{m}\left(\frac{[\tilde{\mathbf{T}}\mathbf{B}\tilde{\mathbf{T}}^\top]_{i,j}}{\tilde{T}_{i,k}B_{k,l}\tilde{T}_{j,l}}T_{i,k}^2B_{k,l}^2T_{j,l}^2 - 2A_{i,j}T_{i,k}B_{k,l}T_{j,l}\right) \\
\leq\ & A_{i,j}^2 + \frac{1}{2}\sum_{k,l=1}^{m}[\tilde{\mathbf{T}}\mathbf{B}\tilde{\mathbf{T}}^\top]_{i,j}\tilde{T}_{i,k}B_{k,l}\tilde{T}_{j,l}\left(\frac{T_{i,k}^4}{\tilde{T}_{i,k}^4} + \frac{T_{j,l}^4}{\tilde{T}_{j,l}^4}\right) \\
& - 2\sum_{k,l=1}^{m} A_{i,j}\tilde{T}_{i,k}B_{k,l}\tilde{T}_{j,l}\left(1 + \log T_{i,k} + \log T_{j,l} - \log\tilde{T}_{i,k} - \log\tilde{T}_{j,l}\right)
\end{aligned}
$$

In the above, $\tilde{\mathbf{T}}$ refers to the matrix $\mathbf{T}$ from the last iteration. We use the convexity of the quadratic function in the first step of the derivation, and the concaveness of the logarithm function in the third step of the derivation. Then, we can upper-bound the first term in the function (2.5) as

$$
\begin{aligned}
&\sum_{i,j=1}^{n}\left(A_{i,j} - \sum_{k,l=1}^{m} T_{i,k}B_{k,l}T_{j,l}\right)^2 \\
\leq\ & \sum_{i,j=1}^{n}\left\{A_{i,j}^2 + \sum_{l=1}^{m}[\tilde{\mathbf{T}}\mathbf{B}\tilde{\mathbf{T}}^\top]_{i,j}[\tilde{\mathbf{T}}\mathbf{B}]_{i,l}\frac{T_{j,l}^4}{\tilde{T}_{j,l}^3} - 4\sum_{l=1}^{m}A_{i,j}[\tilde{\mathbf{T}}\mathbf{B}]_{i,l}\tilde{T}_{j,l}\log T_{j,l}\right. \\
& \left. - 2A_{i,j}[\tilde{\mathbf{T}}\mathbf{B}\tilde{\mathbf{T}}^\top]_{i,j} + 4\sum_{k=1}^{m}A_{i,j}\tilde{T}_{i,k}[\mathbf{B}\tilde{\mathbf{T}}^\top]_{k,j}\log\tilde{T}_{i,k}\right\}
\end{aligned}
$$

Similarly, we can upper-bound the second term in (2.5) as follows:

$$
\begin{aligned}
& C\sum_{j=1}^{n}\sum_{l=1}^{m}(T_{j,l} - \alpha_j\hat{T}_{j,l})^2 \\
=\ & C\sum_{j=1}^{n}\sum_{l=1}^{m}(T_{j,l}^2 - 2\alpha_j\hat{T}_{j,l}T_{j,l} + \alpha_j^2\hat{T}_{j,l}^2) \\
\leq\ & C\sum_{j=1}^{n}\sum_{l=1}^{m}\left[T_{j,l}^2 - 2\alpha_j\hat{T}_{j,l}\tilde{T}_{j,l}(\log\frac{T_{j,l}}{\tilde{T}_{j,l}} + 1) + \alpha_j^2\hat{T}_{j,l}^2\right]
\end{aligned}
$$

By combining the above two bounds, we have the upper bound for the objective function in (2.5). Taking the derivative of the bounding function with respect to $T_{j,l}$, we have

$$4\sum_{i=1}^{n}[\tilde{\mathbf{T}}\mathbf{B}\tilde{\mathbf{T}}^{\top}]_{i,j}[\tilde{\mathbf{T}}\mathbf{B}]_{i,l}\frac{T_{j,l}^{3}}{\tilde{T}_{j,l}^{3}} - 4\sum_{i=1}^{n}A_{i,j}[\tilde{\mathbf{T}}\mathbf{B}]_{i,l}\tilde{T}_{j,l}\frac{1}{T_{j,l}} + C(2T_{j,l} - 2\alpha_j\hat{T}_{j,l}\tilde{T}_{j,l}\frac{1}{T_{j,l}}) = 0$$

which leads to the following solution

$$T_{j,l} = \left[\frac{-C\tilde{T}_{j,l}^{3} + \sqrt{C^2 + 8U_{j,l}\tilde{T}_{j,l}^{4}(2V_{j,l} + C\hat{T}_{j,l}\alpha_j)}}{4U_{j,l}}\right]^{\frac{1}{2}} \tag{2.6}$$

where $U_{j,l} = [\tilde{\mathbf{T}}\mathbf{B}\tilde{\mathbf{T}}^{\top}\tilde{\mathbf{T}}\mathbf{B}]_{j,l}$ and $V_{j,l} = [\mathbf{A}\tilde{\mathbf{T}}\mathbf{B}]_{j,l}$.

In the second step, we fix the unnormalized label confidence $\mathbf{T}_{i,k}$ and search for the normalized label confidence $\tilde{T}_{i,k}$ that optimizes the problem in (2.5), which leads to the following optimal solution:

$$\hat{T}_{j,l} = \frac{T_{j,l}}{\sum_{l=1}^{m}T_{j,l}},\ j = n_l + 1, \ldots, n, l = 1, \ldots, m \tag{2.7}$$

$$\alpha_j = \sum_{l=1}^{m}T_{j,l},\ j = n_l, \ldots, n \tag{2.8}$$

In summary, the iterative steps solving the optimization problem (2.5) could be formulated as a algorithm shown in Table 2.1.

Step 1 Randomly initialize $\mathbf{T}$ and $\hat{\mathbf{T}}$ subject to the constraints in (2.5)
Step 2 Until convergence, do 1. Fix all α_j's and $\hat{\mathbf{T}}$, update $\mathbf{T}$ using Equation (2.6) 2. Fix $\mathbf{T}$, update $\hat{\mathbf{T}}$ using Equation (2.7) 3. Fix $\mathbf{T}$, update all α_j's using Equation (2.8)

Table 2.1. The CNMF algorithm

2.5 Experiments and Discussions

Our experiments are designed to evaluate our proposed multi-label learning framework in text categorization tasks, particularly in the case of a large number of classes

and a small size of training data.

2.5.1 Experiment Setup

The dataset used in our study comes from the textual data of *The Eurovision St Andrews Photographic Collection (ESTA)* in ImageCLEF collection [77]. We randomly pick 3456 documents, and choose the top 100 most popular ones from all the categories those picked documents belong to. On average, each document is assigned to 4.5 classes. Documents are preprocessed by the SMART system with stop words removed and words stemmed, and each document is represented by a vector of weighted term frequency.

Our proposed framework is implemented in the following way. The document similarity $A_{i,j}$ is computed as the cosine similarity between the corresponding term frequency vectors. To compute the class similarity matrix $\mathbf{B}$, we first represent each class c by a binary vector whose elements are set to be one when the corresponding training documents belong to the class c and zero otherwise. We then compute the pairwise class similarity based on their vector representation using a normalized RBF kernel. Finally, the class assignment for each test document is made by the ranking of the label confidence scores that are obtained from the matrix $\mathbf{T}$. Every experiment is repeated 10 times by randomly re-splitting the dataset into the training and the testing sets. The parameter C in the objective function (2.5) is set to 100. We also varied the value of C from 20 to 200, and found that the results remain almost unchanged. For an easy reference, we will refer to the proposed algorithm as "**CNMF**".

Since our approach only produces a ranked list of class labels for a test document, in this study, we focus on evaluating the quality of class ranking. In particular, for each test document, we compute the precision/recall and the $F1$ measurement at each rank by comparing the ranked classes to the true class labels. Then, the

precision/recall and the $F1$ measurement averaged over all the test documents is used as the final evaluation metric.

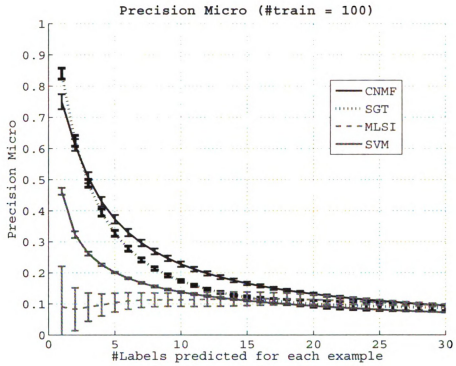
Three baseline models are used in our study. The first one is Spectral Graph Transducer ("SGT" for short) [87], which has been proved effective for exploring unlabeled data. An separated SGT classifier is built for each individual document category, and the probability values output by SGT are used to rank the class labels. The second baseline model is Multi-label Informed Latent Semantic Indexing ("MLSI" for short) [164], which maps document vectors into a low-dimensional space that is strongly correlated with the class labels of the documents. It has been shown empirically that MLSI is effective for exploring both the unlabeled data and the correlation among classes. The last baseline model is Support Vector Machine ("SVM" for short). A linear SVM classifier based on the term frequency vectors of the documents is built for each category independently. All the baseline models are tested by a 10-fold experiment, using the same training/test split of the dataset as the proposed framework.

2.5.2 Experiment Results

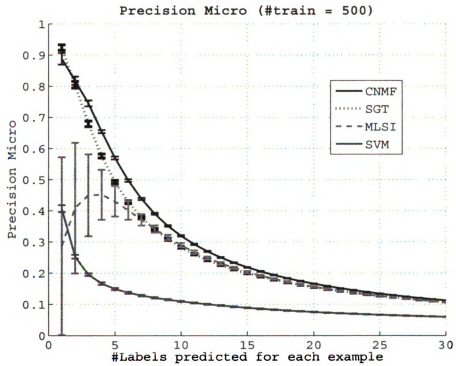
Figure 2.1, Figure 2.2 and Figure 2.3 show the average precision, recall, and $F1$ measurement, respectively, at different ranks, for both the proposed framework and the three baseline approaches.

A comparative analysis based on the results in Figure 2.1, Figure 2.2 and Figure 2.3 lead to the following findings:

1. All four approaches show a same trend of decreasing precision and increasing recall, when the number of labels predicted for each document increases. This is in accordance with the usual precision-recall tradeoff. However, as a measurement balancing the precision and recall, each $F1$ curve clearly shows a peak. As can been seen from Figure 2.3, the $F1$ curve of **CNMF** reaches its climax when

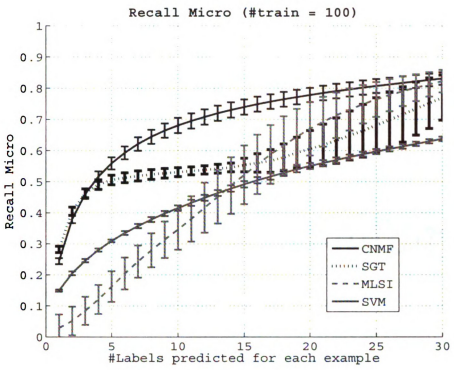


(a) #Training=100

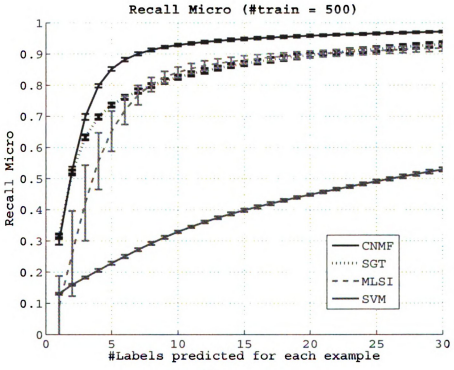


(b) #Training=500

Figure 2.1. Classification performance (Precision) when varying the number of predicted labels for each test example along the ranked list of class labels.

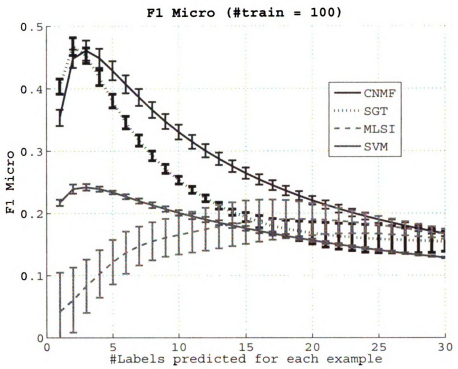


(a) #Training=100

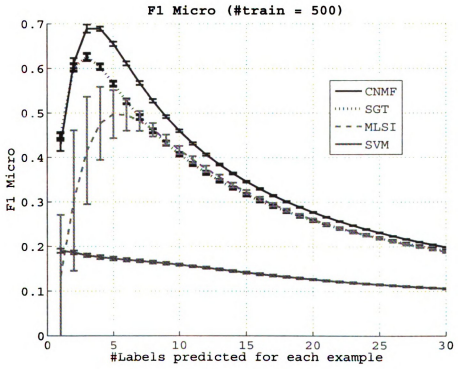


(b) #Training=500

Figure 2.2. Classification performance (Recall) when varying the number of predicted labels for each test example along the ranked list of class labels.



(a) #Training=100



(b) #Training=500

Figure 2.3. Classification performance (F1) when varying the number of predicted labels for each test example along the ranked list of class labels.

the number of predicted labels is around 3 to 4, which is close to the average number of labels per document (i.e., 4.5).

2. **CNMF** makes more significant improvement on the average recall than on the average precision when compared to the three baseline approaches. This is related to our task scenario, which focuses on multi-label learning with a large number of classes and a small size of training examples. Given such a scenario, we would expect a number of classes that are not provided with sufficient amount of training examples. As a result, we hypothesize that prediction on these classes will have to rely heavily on the correlation among classes. This hypothesis is partially justified by the comparison between the proposed approach, **CNMF**, that exploits class correlations, and SGT or SVM, which does not. Although **CNMF** and SGT achieve similar performance in precision, **CNMF** performs significantly better than the SGT in terms of the average recall.

3. More improvement by **CNMF** to the three baseline approaches is observed when the number of training documents is 100 than when the number of training documents is 500. This is partially due to the same reason mentioned above – the advantage of exploiting class correlations on sparse training data. It can also be attributed to the reason that our approach also makes use of the correlation among the unlabeled data, which has been proved by many studies in semi-supervised learning, for instance [173, 129, 172].

4. Although MLSI is able to explore the correlation among classes, its performance depends heavily on the appropriate choice of the number of latent variables and the tuning parameter determining how much the indexing should be biased by the outputs. These two parameters are usually determined by a cross validation approach and therefore could be problematic when the number of training examples is relatively small. This problem is directly reflected in the large vari-

ance in both precision and recall of the MLSI algorithm, which we believe it is due to the inappropriate choice of the aforementioned two parameters given the limited number of training examples.

2.6 Conclusions

In this chapter, we propose a novel framework to accommodate the side information of class correlation in multi-label learning, which meets the challenging situation of a large number of classes and a small size of training data. The advantage of our proposed framework is that it is able to exploit the correlation among classes and the unlabeled data. We also present an efficient algorithm to solve the related optimization problem. Experiments show that our proposed framework performs significantly better than the other three state-of-the-art multi-label learning techniques in text classification tasks.

CHAPTER 3

Semi-supervised Classification with Link Information

3.1 Introduction

From supervised learning to semi-supervised learning, machine learning research has witnessed the trend of exploiting the structure of unlabeled data. A typical approach towards revealing the underlying structure of unlabeled data is through establishing correlation between example pairs from their data representation (or feature values). For example, many graph-based learning models construct a k-nearest neighbor graph by choosing an appropriate distance measurement defined on the data representation of a pair of examples, such as Local Linear Embedding [124], ISOMap [142], Laplacian Eigenmaps [17], Manifold Learning [18] and etc.. However, many real-world datasets already exhibit inherent correlations by entities that are interlinked with each other. Especially in text domains, datasets features with "links" are very popular: nearly all kinds of scientific publications are cross-referenced by each other; and the World Wide Web is weaved by hyperlinks that connect pages. One can also find datasets with link information in other domains, such as social networks, bioinformatics, etc.

Naturally, link information suggests certain structure underlying the dataset. For example, an empirical study showed that the Web's spatial locality (induced by hy-

perlinks) mirrors its topical locality [47]. Moreover, link information usually provides a different view on the structure of data than that "computed" from the data representations, since link generation often involves human interference. For example, when a web author constructs hyperlinks, he or she can capture more sophisticated semantic closeness between web pages that is beyond the power of state-of-the-art text mining tools. Therefore, exploiting link information leaves the hope of gaining more insight into the structure of data for many learning algorithms.

However, while it could be informative, link information also often presents some annoying characteristics in practice. First, link information could be sparse, i.e., some data examples could be involved in no links. Second, link information are often incomplete, i.e., one cannot always expect a link being observed wherever a link "should" be constructed between two examples. For example, we cannot hope a scientific publication to cite all work that is related to itself. Third, link information tends to be noisy. One example to illustrate noise in links is the hyperlink spam on the Web. In summary, in real-world datasets, link information is typically not a reliable source for the structure of data. Therefore, link information is always treated as "side information" that supplements the original data representations for learning tasks.

Informative but unreliable, link information provides opportunity as well as challenge for semi-supervised learning. In this chapter, we will focus on classification on datasets with link structures. Depending on how link information is incorporated into a learning algorithm, previous studies on this topic can be roughly categorized into three types of models: representation augmentation, correlation augmentation and training pool augmentation. The first type modifies the representation of a data example from its neighborhood that is induced from the link structure [33, 118, 61, 63, 162, 48, 120, 130, 40]. The second type utilizes link information to construct a graph [168, 167, 7], or derive more accurate similarity measurement

between data examples [32], and apply (or design) semi-supervised learning algorithm over the graph. Essentially, link information plays the role of establishing correlation between examples. The third type uses linked examples to augment the training set, represented by the co-training algorithm and its variants [27, 39]. Usually a feature based classifier and a link based classifier are applied alternatively, supplementing each other's training pool with its highly confident predictions.

However, the performance of all three types of models summarized above can degrade significantly with a decreasing number of links. When links are sparse, only a small fraction of data is involved in some link(s). No matter a model augments the data representation, correlation or training pool, an unbalanced issue is created between those data examples that are influenced by the links and the rest that are not. To overcome this problem brought by the sparseness of link information, it is desirable that the limited link information can be somehow "smoothed" over the entire dataset.

In this chapter, we will propose a novel semi-supervised framework, termed as "**LinkBoost**", to improve classification accuracy by utilizing link information. LinkBoost is a boosting framework: a base classifier is applied iteratively with augmented training pool, which is updated through a minimization of inconsistency between the class label assignments and the pairwise data similarities that is augmented by the link information; with a series of learned weights, the classification results from all iterations are finally combined as the final prediction. Thus the classification results from the iteratively applied classifier with updating training pool act as a way of propagating link information around, i.e., "smoothing" the influence of links over the entire dataset. As a result, LinkBoost is more robust against the sparseness of link information. Besides, as a general boosting framework, LinkBoost can be applied to *any* classification algorithm. LinkBoost framework is also hybrid in the sense that it can accommodate all three augmentations from link information on the dataset: rep-

resentation augmentation, correlation augmentation and training pool augmentation.

The following notations will be used throughout this chapter. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ denote a set of n examples, where the first n_l ones are labeled and the rest n_u ones are unlabeled, i.e. $n = n_l + n_u$. We also use $\mathbf{x}_i$ to represent the feature vector of the i-th example, and use $\mathbf{X}$ to denote the matrix that gathers all the feature vectors of examples, i.e., $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$. Let $\mathbf{y} = \{y_i\}_{i=1}^{n}$ be the label vector, in which $\{y_i\}_{i=1}^{n_l}$ is given, and $\{y_i\}_{i=n_l+1}^{n_l+n_u}$ is to be decided. $s_{i,j}$ is defined as a similarity measurement between the i-th and the j-th examples, and a matrix is formed as $\mathbf{S} = [s_{i,j}]_{n\times n}$. The link information is encoded into a matrix $\mathbf{R} = [r_{i,j}]_{n\times n}$, where $r_{i,j} = 1$ if there is a link from $\mathbf{x}_i$ to $\mathbf{x}_j$ and $r_{i,j} = 0$ otherwise. From the link matrix, a co-citation matrix can be defined as $\mathbf{C} = [c_{i,j}]_{n\times n}$, where $c_{i,j} = 1$ if $\exists k \neq i, j$ such that $r_{i,k} = 1, r_{j,k} = 1$, and $c_{i,j} = 0$ otherwise [1].

3.2 Related Work

In this section, we will briefly review previous efforts in the area of link-based classification (especially in text domain), and a few semi-supervised classification models that are closely related to this chapter, followed by a recapitulation of several representative algorithms.

3.2.1 Link-based Classification

The most popular way of link-based classification is to augment the representation of data examples with their neighbors. In [63, 32, 120, 130], the bag-of-words model of documents is augmented by including words from neighboring documents, thus creating a "virtual document" to feed into classifiers. However, the studies in [33, 118]

[1]This co-citation matrix is defined in the "co-citing" manner, i.e., two examples are correlated if they both link *to* a third example. Another way is to define co-citation matrix in a "co-cited" manner, i.e., two examples are correlated if they are both linked *from* a third example. Deciding which definition is more appropriate is domain dependent.

suggests that incorporating words from neighboring documents sometimes leads to performance degradation, while making use of their predicted class labels as additional features for data representation was helpful. Later in [107], different schemes of formulating the additional link-based features were discussed. Apart from explicitly augmenting data representations with features or labels of linked examples, another stream of research tries to unify content analysis with link analysis by creating new data representation for linked data examples in the latent space [48, 40].

When links are often directional (which is true in most cases), there exists several different criteria to identify neighborhood for a data example. The most frequently used criteria include incoming links, outgoing links and co-citation links [107, 61, 162, 32]. Especially, the studies presented in [61, 162] suggests that different dataset regularities may favor the use of different neighborhood identification criteria.

Rather than augmenting data representation to better explore the underlying structure of data for semi-supervised learning, a few studies directly create a graph structure from link information. These approaches often leads to semi-supervised learning algorithms over graphs: an iterative relaxation labeling algorithm is proposed in [7] for undirected graph; the work in [168, 167] handles directed graph by regularizing classification functions to change slowly on densely linked subgraphs.

3.2.2 Related Semi-supervised Classification Models

A large number of models have been proposed for semi-supervised classification, which can be broadly categorized into several types: graph-based models, margin-based models, kernel-based models, ensemble-based models, and etc. Although only a small fraction of models directly address the use of link information, at least two types of models are closely related to the LinkBoost framework proposed in this chapter: graph-based models and ensemble-based models.

Graph-based models build a connected graph on both labeled and unlabeled ex-

amples, with each vertex representing an example and each weighted edge representing correlations between example pairs. The most well-known graph-based models include Harmonic Functions [173, 174], Spectral Graph Transducer [87], Gaussian Processes [4, 103, 67], Manifold Regularization [18], Label Propagation [19], etc. A common theme shared by many graph-based semi-supervised classification models is to find an optimal set of class labels for unlabeled examples, such that they are consistent with supervised class labels from labeled examples, as well as the graph structure. Graph Laplacian is a popular form to define an inconsistency measurement over a set of class assignment $\mathbf{y} = \{y_i\}$ and the graph represented by a similarity matrix $\mathbf{S} = [s_{i,j}]$

$$F(\mathbf{y}) = \sum_{i=1}^{n} \sum_{j=1}^{n} s_{i,j}(y_i - y_j)^2 = \mathbf{y}^\top \mathbf{L}\mathbf{y}$$

he above inconsistency measurement, is always combined with other components to form an objective function for minimization. It is easy to imagine, when link information is available, we can enforce class assignment to be consist with it, in a similar way as graph-based models. As will be seen in Section 3.3, the proposed LinkBoost framework uses a similar definition inconsistency measurement, except in the form of exponential cost (to facilitate deriving a boosting algorithm) rather than quadratic cost. Moreover, unlike most graph-based models, which are non-parametric and do not build specific classification models, the proposed LinkBoost framework is able to yield one classification model based on *any* given classification algorithm. This is particularly useful, when the amount of unlabeled data is extremely large.

Ensemble-based models gained increasing attention in semi-supervised learning by several successful models, such as AdaBoost [56, 55], ASSEMBLE [21] and Semi-supervised Margin Boost (SSMB) [43]. Usually, in an iterative manner, pseudo labels are assigned to unlabeled examples, which are then sampled for training a new supervised classifier from an ensemble of them. The proposed LinkBoost framework follows the idea of iteratively augmenting the training set, but it utilizes link infor-

mation to obtain more reliable pseudo labels for unlabeled examples. In this sense, LinkBoost framework combines the advantages from both graph-based models and ensemble-based models.

3.2.3 Representative Algorithms Review

Nearly all the algorithms proposed on link-based classification make the assumption that interlinked (or co-citation) examples are more likely to belong to the same class. In the following, we will recapitulate a few representative algorithms proposed for semi-supervised classification on datasets with link information.

Feature Representation Augmentation

In the family of feature representation augmentation algorithms, a example's feature set is supplemented with features from its interlinked/co-citation examples. Formally, we can write the augmented example as

$$\mathbf{X}^{aug} = (1-\lambda)\mathbf{X} + \lambda\mathbf{M}^{\top}\mathbf{X} \tag{3.1}$$

where the matrix $\mathbf{M} = \mathbf{R}$ if we only want to augment an example with incoming links, or $\mathbf{M} = \mathbf{R} + \mathbf{R}^{\top}$ if both incoming links and outgoing links are used, or $\mathbf{M} = \mathbf{C}$ if only co-citation examples are considered for augmentation. λ is a weight to put on the augmentation from linked examples.

The augmented feature representation will be fed into an semi-supervised learning algorithm for classification.

Co-training for Link-based Classification

Co-training is first proposed in [27], in which two or possibly more learners are trained separately on a set of examples; each learner uses a different, and ideally independent, set of features for each example. Co-training can be naturally applied for link-based

classification: one learner is based on the data representation, the other learner is based on link information.

Specifically, the co-training algorithm for link-based classification can be formalized as follows

Co-training for Link-based Classification

Step 1 Initialize a training pool with all training examples $\mathbf{R} = \mathbf{x}_1, \cdots, \mathbf{x}_{n_l}$.

Step 2 Iteratively apply the following two learners

- Train a data representation based learner on current training pool $\mathbf{R}$, and apply the learner to predict labels for the rest examples. Update the training pool $\mathbf{R}$ by move those examples with high prediction confidence into it.
- Train a link based learner on current training pool $\mathbf{R}$, and apply the learner to predict labels for the rest examples. Update the training pool $\mathbf{R}$ by move those examples with high prediction confidence into it.

Until no more examples can be added into the training pool $\mathbf{T}$.

Iterative Classification Algorithm

Lu & Getoor proposed an iterative classification Algorithm (ICA) in [107]. Different from the feature representation augmentation approach, ICA augments the data representation by a new set of features that summarize the class label statistics (from prediction of the previous iteration) of interlinked/co-citation examples. Due to the fact that newly introduced label feature set is different from the original feature set in nature, two logistic regression models on both feature sets are combined to form a prediction.

To formalize the ICA algorithm, let us introduce $\mathbf{z}_i$ to represent the new feature vector, i.e., linked examples' label statistics, for the corresponding data example

$\mathbf{x}_i$. For example, for a C-way classification problem, $\mathbf{z}_i$ could be defined as a C-dimensional vector, where a element $z_{i,k}$ counts the number of examples from those linked with $\mathbf{x}_i$ that belong to the k-th class.

Then the ICA algorithm can be summarized as follows

Iterative Classification Algorithm

(Bootstrap) Initially assign class to each example based on its original feature representation $\mathbf{x}_i$.

(Iteration) Iteratively apply the following two learners

- Form a new feature representation $\mathbf{z}_i$ for each example $\mathbf{x}_i$, by gathering the label statistics based on current class assignment to linked examples.
- Train two separate logistic regression models on both feature representations, i.e., $\mathbf{x}_i$ and $\mathbf{z}_i$, by the following MAP estimation

$$\Pr(\mathbf{y}|\mathbf{X}) = \Pr(\mathbf{y}|\{\mathbf{x}_i\})\Pr(\mathbf{y}|\{\mathbf{z}_i\})$$

where

$$\Pr(\mathbf{y}|\{\mathbf{x}_i\}) = \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{\exp(-y_i \mathbf{w}_x^\top \mathbf{x}_i) + 1}$$

$$\Pr(\mathbf{y}|\{\mathbf{z}_i\}) = \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{\exp(-y_i \mathbf{w}_z^\top \mathbf{z}_i) + 1}$$

In the above, $\mathbf{w}_x$ and $\mathbf{w}_z$ are two parameters for the logistic regression model on either feature set.

- Apply the combined logistic regression model to the test examples and update their class assignments.

Until no updates are made on class assignments or a maximum number iterations has been reached.

3.3 LinkBoost Framework

For simplicity, here we only consider binary classification, where the label vector $\mathbf{y}$ can only take values $y_i = 1$ or $y_i = -1$.

As a semi-supervised framework, LinkBoost inherits the common theme from graph-based models that consistency is enforced between the link-information induced graph structure and the class assignment to the unlabeled data. On the other hand, as a general framework that is able to boost *any* base supervise learning algorithm, LinkBoost follows the idea of ensemble-based models that a sampling of unlabeled examples based on predicted pseudo labels will be used to iteratively augment the training set of the specified supervised algorithm. In the rest of the section, we will formalize the LinkBoost framework and derive the related algorithm.

3.3.1 Objective Function

To encode the structure of both labeled and unlabeled data, we use a similarity matrix $\mathbf{S} = [s_{i,j}]_{n \times n}$ to combine both the link information and the feature representation of data examples

$$s_{i,j} = (1 - \alpha) sim(\mathbf{x}_i, \mathbf{x}_j) + \alpha r_{i,j} \tag{3.2}$$

Here $sim(\cdot, \cdot)$ defines a similarity measurement based on the features. For example, we can use cosine similarity in text domain. $r_{i,j}$ is an element from the link matrix $\mathbf{R}$ which, more specifically, encodes all the incoming links; the link matrix $\mathbf{R}$ can be replaced with $\mathbf{R}+\mathbf{R}^\top$ if both incoming and outgoing links are taken into consideration, or be replaced by the co-citation matrix $\mathbf{C}$ if co-citation links are more appropriate to disclose the structure of data. Making a good choice on the link matrix is usually dependent on the regularity presented by the dataset for classification, as suggested in [61, 162]. α is a combination weight factor, $0 \leq \alpha \leq 1$. The above similarity definition can be viewed as using data represenatino based similarity computation to

smooth the links, thus minizing the problem brought by the sparseness in the link information.

We define the inconsistency, *within the unlabeled data*, between the class labels $\{y_i\}_{i=n_l+1}^{n_l+n_u}$ and the similarity measurement $\mathbf{S}$ as

$$F_{uu} = \sum_{i,j=n_l+1}^{n_l+n_u} s_{i,j} \cosh(y_i - y_j) \tag{3.3}$$

where $\cosh(\cdot)$ is the hyperbolic cosine function, i.e., $\cosh(y_i - y_j) = [\exp(y_i - y_j) + \exp(y_j - y_i)]/2$. When symmetric similarity measurement is considered, as in this paper, the above inconsistency can be rewritten as

$$\begin{aligned} F_{uu} &= \sum_{i,j=n_l+1}^{n_l+n_u} \frac{1}{2} s_{i,j} \exp(y_i - y_j) + \sum_{i,j=n_l+1}^{n_l+n_u} \frac{1}{2} s_{j,i} \exp(y_j - y_i) \\ &= \sum_{i,j=n_l+1}^{n_l+n_u} s_{i,j} \exp(y_i - y_j) \end{aligned} \tag{3.4}$$

Also, we define the inconsistency, *across the labeled and unlabeled data*, between the class labels and the similarity measurement as

$$F_{lu} = \sum_{i=1}^{n_l} \sum_{j=n_l+1}^{n_l+n_u} s_{i,j} \exp(-2y_i y_j) \tag{3.5}$$

Ideally, the labels decided for the unlabeled data should minimize both inconsistencies stated above. This leads to the following optimization problem

$$\min_{\{y_i, i=n_l+1,\ldots,n_l+n_u\}} F_{uu} F_{lu} \tag{3.6}$$

3.3.2 Boosting Algorithm

In this subsection, we will derive a boosting algorithm that solves the optimization problem (3.6) in an iterative procedure. Specifically, given an arbitrary binary classification algorithm $\mathcal{A}$, let $h^{(t)}(\mathbf{x})$ denote the classification model learned in the t-th

iteration by this algorithm. Then the final classification model, after T iterations, is the combination of T models learned at all iterations, i.e.

$$H^{(T)}(\mathbf{x}) = \sum_{t=1}^{T} \alpha^{(t)} h^{(t)}(\mathbf{x}) \tag{3.7}$$

where the $\alpha^{(t)} \geq 0$ is the combination weight. Here the superscript in parenthesis indicates the iteration number. Finally we will apply this model to predict the labels for the unlabeled data, i.e., $y_i = H^{(T)}(\mathbf{x}_i), i = n_l + 1, \ldots, n_l + n_u$. To derive a boosting procedure that minimize the objective function (3.6), we need to find a good combination weight at each iteration, so that the objection function yields a decreased value from previous iteration.

Now we study the change of objective function from the $t-1$ iteration to the t-th iteration. To simplify the notation, let H_i denote the class label of the i-th example (unlabeled) predicted by the combined model from all $t-1$ iterations, and h_i denote the same example's predicted label at the t-th iteration. We also simplify $\alpha^{(t)}$ as α. Now the objective function becomes

$$\begin{aligned} F^{(t)} &= F_{uu}^{(t)} F_{lu}^{(t)} \\ &= \left[\sum_{i,j=n_l+1}^{n_l+n_u} s_{i,j} \exp(H_i + \alpha h_i - H_j - \alpha h_j) \right] \\ &\quad \cdot \left[\sum_{i=1}^{n_l} \sum_{j=n_l+1}^{n_l+n_u} s_{i,j} \exp(-2y_i(H_j + \alpha h_j)) \right] \end{aligned} \tag{3.8}$$

Using the inequality of arithmetic and geometric means, we have

$$\exp[\alpha(h_i - h_j)] \leq \frac{1}{2}\left[\exp(2\alpha h_i) + \exp(-2\alpha h_j)\right] \tag{3.9}$$

So we can bound the first term in (3.8), $F_{uu}^{(t)}$, as follows

$$\begin{aligned}
F_{uu}^{(t)} &= \sum_{i,j=n_l+1}^{n_l+n_u} s_{i,j} \exp(H_i + \alpha h_i - H_j - \alpha h_j) \\
&= \sum_{i,j=n_l+1}^{n_l+n_u} s_{i,j} \exp(H_i - H_j) \exp(\alpha(h_i - h_j)) \\
&\leq \sum_{i,j=n_l+1}^{n_l+n_u} \frac{1}{2} s_{i,j} \exp(H_i - H_j) \left[\exp(2\alpha h_i) + \exp(-2\alpha h_j)\right] \\
&= \sum_{i=n_l+1}^{n_l+n_u} \exp(2\alpha h_i) \sum_{j=n_l+1}^{n_l+n_u} \frac{1}{2} s_{i,j} \exp(H_i - H_j) \\
&\quad + \sum_{j=n_l+1}^{n_l+n_u} \exp(-2\alpha h_j) \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{2} s_{i,j} \exp(H_i - H_j) \\
&= \sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j) \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{2} s_{j,i} \exp(H_j - H_i) \\
&\quad + \sum_{j=n_l+1}^{n_l+n_u} \exp(-2\alpha h_j) \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{2} s_{i,j} \exp(H_i - H_j) \qquad (3.10)
\end{aligned}$$

In the last step above, we switched the index i and j in the first term for notational convenience.

If we define

$$a_j \stackrel{\text{def}}{=} \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{2} s_{j,i} \exp(H_j - H_i) \qquad (3.11)$$

$$b_j \stackrel{\text{def}}{=} \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{2} s_{i,j} \exp(H_i - H_j) \qquad (3.12)$$

then we can simplify the bound for the $F_{uu}^{(t)}$ in (3.10) as

$$F_{uu}^{(t)} \leq \sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j) a_j + \exp(-2\alpha h_j) b_j \qquad (3.13)$$

Similarly, we can simplify $F_{uu}^{(t-1)}$ as

$$\begin{aligned}
F_{uu}^{(t-1)} &= \sum_{i,j=n_l+1}^{n_l+n_u} s_{i,j} \exp(H_i - H_j) \\
&= \sum_{i=n_l+1}^{n_l+n_u} \sum_{j=n_l+1}^{n_l+n_u} \frac{1}{2} s_{i,j} \exp(H_i - H_j) + \sum_{j=n_l+1}^{n_l+n_u} \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{2} s_{i,j} \exp(H_i - H_j) \\
&= \sum_{j=n_l+1}^{n_l+n_u} \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{2} s_{j,i} \exp(H_j - H_i) + \sum_{j=n_l+1}^{n_l+n_u} \sum_{i=n_l+1}^{n_l+n_u} \frac{1}{2} s_{i,j} \exp(H_i - H_j) \\
&= \sum_{j=n_l+1}^{n_l+n_u} a_j + b_j \qquad (3.14)
\end{aligned}$$

The second term in (3.8), $F_{lu}^{(t)}$, can be rewritten as

$$\begin{aligned}
F_{lu}^{(t)} &= \sum_{i=1}^{n_l} \sum_{j=n_l+1}^{n_l+n_u} s_{i,j} \exp(-2y_i(H_j + \alpha h_j)) \\
&= \sum_{j=n_l+1}^{n_l+n_u} \sum_{i=1}^{n_l} s_{i,j} \exp(-2y_i H_j) \exp(-2\alpha y_i h_j) \\
&= \sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j) \sum_{i=1}^{n_l} s_{i,j} \exp(2H_j) \delta(y_i, -1) \\
&\quad + \sum_{j=n_l+1}^{n_l+n_u} \exp(-2\alpha h_j) \sum_{i=1}^{n_l} s_{i,j} \exp(-2H_j) \delta(y_i, 1) \qquad (3.15)
\end{aligned}$$

In the above, we use the fact that the label for a labeled example, y_i, can only be 1 or -1. And the delta function $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ if $x \neq y$.

Again, if we define

$$c_j \stackrel{\text{def}}{=} \sum_{i=1}^{n_l} s_{i,j} \exp(2H_j) \delta(y_i, -1) \qquad (3.16)$$

$$d_j \stackrel{\text{def}}{=} \sum_{i=1}^{n_l} s_{i,j} \exp(-2H_j) \delta(y_i, 1) \qquad (3.17)$$

then we can simplify the bound for the $F_{uu}^{(t)}$ in (3.10) as

$$F_{uu}^{(t)} = \sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j)c_j + \exp(-2\alpha h_j)d_j \tag{3.18}$$

Similarly, we can simplify $F_{lu}^{(t-1)}$ as

$$\begin{aligned} F_{lu}^{(t-1)} &= \sum_{i=1}^{n_l} \sum_{j=n_l+1}^{n_l+n_u} s_{i,j} \exp(-2y_i H_j) \\ &= \sum_{j=n_l+1}^{n_l+n_u} \sum_{i=1}^{n_l} s_{i,j} \exp(2H_j)\delta(y_i, -1) + \exp(-2H_j)\delta(y_i, 1) \\ &= \sum_{j=n_l+1}^{n_l+n_u} c_j + d_j \end{aligned} \tag{3.19}$$

Given the above results, we can study the bound of the objective function change from the $t-1$-th iteration to the t-th iteration. However, to further simplify notations, let us first define

$$\tilde{a}_j = \frac{a_j}{\sum_{j=n_l+1}^{n_l+n_u} a_j + b_j} \tag{3.20}$$

$$\tilde{b}_j = \frac{b_j}{\sum_{j=n_l+1}^{n_l+n_u} a_j + b_j} \tag{3.21}$$

$$\tilde{c}_j = \frac{c_j}{\sum_{j=n_l+1}^{n_l+n_u} c_j + d_j} \tag{3.22}$$

$$\tilde{d}_j = \frac{d_j}{\sum_{j=n_l+1}^{n_l+n_u} c_j + d_j} \tag{3.23}$$

then we have

$$\begin{aligned}
\log \frac{F^{(t)}}{F^{(t-1)}} &= \log \frac{F_{uu}^{(t)}}{F_{uu}^{(t-1)}} + \log \frac{F_{lu}^{(t)}}{F_{lu}^{(t-1)}} \\
&\leq \log \frac{\sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j) a_j + \exp(-2\alpha h_j) b_j}{\sum_{j=n_l+1}^{n_l+n_u} a_j + b_j} \\
&\quad + \log \frac{\sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j) c_j + \exp(-2\alpha h_j) d_j}{\sum_{j=n_l+1}^{n_l+n_u} c_j + d_j} \\
&= \log \sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j)\tilde{a}_j + \exp(-2\alpha h_j)\tilde{b}_j \\
&\quad + \log \sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j)\tilde{c}_j + \exp(-2\alpha h_j)\tilde{d}_j \\
&\leq \sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j)(\tilde{a}_j + \tilde{c}_j) + \exp(-2\alpha h_j)(\tilde{b}_j + \tilde{d}_j) - 2 \qquad (3.24)
\end{aligned}$$

In the last step above, we used the inequality

$$\log x \leq x - 1, \quad \forall x > 0 \qquad (3.25)$$

Furthermore, if we apply the following inequality

$$\exp(\gamma x) \leq \exp(\gamma) + \exp(-\gamma) - 1 + \gamma x, \quad \forall x \in [-1, +1] \qquad (3.26)$$

we can further bound $\log \frac{F^{(t)}}{F^{(t-1)}}$ as

$$\begin{aligned}
\log \frac{F^{(t)}}{F^{(t-1)}} &\leq \sum_{j=n_l+1}^{n_l+n_u} (\tilde{a}_j + \tilde{c}_j) \left[\exp(2\alpha) + \exp(-2\alpha) - 1 + 2\alpha h_j\right] \\
&\quad + (\tilde{b}_j + \tilde{d}_j) \left[\exp(2\alpha) + \exp(-2\alpha) - 1 - 2\alpha h_j\right] \\
&= \sum_{j=n_l+1}^{n_l+n_u} (\tilde{a}_j + \tilde{b}_j + \tilde{c}_j + \tilde{d}_j) \left[\exp(2\alpha) + \exp(-2\alpha) - 1\right] \\
&\quad - \sum_{j=n_l+1}^{n_l+n_u} 2\alpha h_j (\tilde{b}_j + \tilde{d}_j - \tilde{a}_j - \tilde{c}_j) - 2 \qquad (3.27)
\end{aligned}$$

Now we have two upper bounds for $\log \frac{F^{(t)}}{F^{(t-1)}}$, (3.24) and (3.27). Both bounds and $\log \frac{F^{(t)}}{F^{(t-1)}}$ "touch" at $\alpha = 0$, since they all reach 0 when $\alpha = 0$. Furthermore, $\log \frac{F^{(t)}}{F^{(t-1)}} = 0$ means $F^{(t)} = F^{(t-1)}$. Therefore, as long as we minimize either upper bound, we can always have $F^{(t)} \leq F^{(t-1)}$, i.e., keeping the objective function non-increasing in the iterative procedures.

The upper bound (3.27) is good for designing a sampling scheme for a boosting algorithm. Specifically, to help lower such an upper bound, we expect the label value h_j (at the t iteration) to be consistent with the sign of $(\tilde{b}_j + \tilde{d}_j - \tilde{a}_j - \tilde{c}_j)$. This gives us a good hint on sampling unlabeled data for training: the j-th unlabeled example should be labeled as $\text{sign}(\tilde{b}_j + \tilde{d}_j - \tilde{a}_j - \tilde{c}_j)$ and sampled with a probability proportional to $|(\tilde{b}_j + \tilde{d}_j - \tilde{a}_j - \tilde{c}_j)|$.

Finally, to find the optimal α, we can minimize either upper bound in (3.24) or (3.27). Here we choose the former, because it is tighter than the other one, and also minimizing it leads to a solution which is easier to compute. In particular, we take the first-order derivative (w.r.t. α) of the upper bound in (3.24) and set it to zero, i.e.

$$\sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha h_j) 2h_j(\tilde{a}_j + \tilde{c}_j) - \sum_{j=n_l+1}^{n_l+n_u} \exp(-2\alpha h_j) 2h_j(\tilde{b}_j + \tilde{d}_j) = 0 \quad (3.28)$$

or equivalently

$$\begin{aligned} &\sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha)(\tilde{a}_j + \tilde{c}_j)\delta(h_j, 1) - \sum_{j=n_l+1}^{n_l+n_u} \exp(-2\alpha)(\tilde{a}_j + \tilde{c}_j)\delta(h_j, -1) \\ &- \sum_{j=n_l+1}^{n_l+n_u} \exp(-2\alpha)(\tilde{b}_j + \tilde{d}_j)\delta(h_j, 1) + \sum_{j=n_l+1}^{n_l+n_u} \exp(2\alpha)(\tilde{b}_j + \tilde{d}_j)\delta(h_j, -1) \\ &= 0 \quad (3.29) \end{aligned}$$

Solving the above equation, we have

$$\alpha = \frac{1}{4} \ln \frac{\sum_{j=n_l+1}^{n_l+n_u} (\tilde{a}_j + \tilde{c}_j)\delta(h_j, -1) + (\tilde{b}_j + \tilde{d}_j)\delta(h_j, 1)}{\sum_{j=n_l+1}^{n_l+n_u} (\tilde{a}_j + \tilde{c}_j)\delta(h_j, 1) + (\tilde{b}_j + \tilde{d}_j)\delta(h_j, -1)} \quad (3.30)$$

Input

- X: $d \times n$ matrix for the input data
- $\mathcal{A}$: given classification algorithm

Output: class labels

Algorithm

- Compute pairwise similarity matrix $\mathbf{S}$.
- Initialize $H^{(0)}(\mathbf{x}) = 0$.
- For $t = 1, 2, \ldots, T$
  - Compute $\tilde{a}_j, \tilde{b}_j, \tilde{c}_j, \tilde{d}_j$ using (3.20)-(3.23)
  - Compute class label for each unlabeled example $\mathbf{x}_j$ as $\text{sign}(\tilde{b}_j + \tilde{d}_j - \tilde{a}_j - \tilde{c}_j)$.
  - Sample unlabeled examples with probability proportional to $|(\tilde{b}_j + \tilde{d}_j - \tilde{a}_j - \tilde{c}_j)|$
  - Adding sampled examples to the labeled examples, train a binary classifier $h^{(t)}(\mathbf{x})$ with the algorithm $\mathcal{A}$
  - Compute the optimal $\alpha^{(t)}$ using (3.30)
  - Update the classification model as $H^{(t)}(\mathbf{x}) \leftarrow H^{(t-1)}(\mathbf{x}) + \alpha^{(t)} h^{(t)}(\mathbf{x})$
- Predict class labels with the final classification model $H^{(t)}(\mathbf{x})$

Figure 3.1. LinkBoost framework.

3.3.3 LinkBoost Framework Summary

The LinkBoost framework can be summarized into a meta algorithm, as shown in Figure 3.1

As we can see, LinkBoost framework is able to take *any* supervised algorithm as the base classifier. This is more useful as opposed to other link-based classification method that design a special algorithm. Consider the situation that we find a specified classifier which works particularly well for a given domain. When we gained more understanding on the dataset (in the form of "links"), we want to stick with

the good classifier and further improve its accuracy. This requires a framework that is flexible enough to accommodate *any* classifier, and is able to evaluate its performance (without knowing ground truth) and make adjustment. LinkBoost meets such a requirement in the following way: it iteratively applies the classifier with a different training set; by checking the inconsistency between current classification results and the structure of data that is estimated from the link information and data representations, it generates heuristics on assigning pseudo labels to unlabeled data, thus updating the training set; at the same time, LinkBoost estimated the appropriate amount of trust we can put on the current prediction for final combination.

It is also worth noting that even when no link information is available, LinkBoost can also work as a generic semi-supervised learning algorithm, by simply using the similarity computed from data representations alone in (3.2) in Section 3.3.1.

3.4 Experiments and Analysis

In this section, we will present a few experiments to verify the proposed LinkBoost framework. Specifically, through the experiments we try to address the following two research questions

1. Is the propose LinkBoost framework effective in improving classification performance with link information?

2. As a general semi-supervised boosting framework, is LinkBoost able to improve *any* supervised classification algorithm?

3.4.1 Experiment Setup

Two datasets of scientific publications will be used in our experiments: "Cora" and "Citeseer". We describe the two datasets here.

ID	Class Name	Size
1	Neural Networks	818
2	Rule Learning	180
3	Reinforcement Learning	217
4	Probabilistic Methods	426
5	Theory	351
6	Genetic Algorithms	418
7	Case Based	298

Table 3.1. Classes in Cora dataset.

ID	Class Name	Size
1	Agents	596
2	Information Retrieval	668
3	Database	701
4	Artificial Intelligence	249
5	Human-computer Interaction	508
6	Machine Learning	590

Table 3.2. Classes in Citeseer dataset.

Cora dataset The Cora dataset consists of 2708 scientific publications classified into one of seven classes, as shown in Table 3.1. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary, which consists of 1433 unique words in total. The citation network consists of 5429 links. By checking with the ground truth class assignment of publications, we find that about 81.38% of the links correctly indicates that the linked publication pair should be put in the same class. Furthermore, the links are not evenly distributed across all publications. For example, there are about 100 publications each involved in more than 10 links, while there are also 1143 publications not involved in any link.

Citeseer dataset The CiteSeer dataset consists of 3312 scientific publications classified into one of six classes, as shown in Table 3.2. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the

corresponding word from the dictionary, which consists of 3703 unique words in total. The citation network consists of 4732 links. By checking with the ground truth class assignment of publications, we find that about 74.61% of the links correctly indicates that the linked publication pair should be put in the same class. Furthermore, the links are not evenly distributed across all publications. For example, there are about 50 publications each involved in more than 10 links, while there are also 1358 publications not involved in any link.

As we can see, the link information in both datasets are sparse and noisy. Therefore, these two datasets can be seen as representative of real-world data.

Given a supervised classification algorithm, we will apply the proposed LinkBoost framework to boost its classification performance on the two datasets described above, using the link information. Due to the regularity presented in the two datasets, we will use both incoming and outgoing links to form a link matrix, i.e., in Equation (3.2) $\mathbf{R}+\mathbf{R}^\top$ will be used as the link matrix to compute similarity between data examples. And the weight factor α is set to 0.4. As mentioned before, the proposed LinkBoost Framework can accommodate augmented feature representation, for example, from the Feature Representation Augmentation algorithm reviewed in Section 3.2.3. In our experiments, we implemented LinkBoost on both the original feature space, and the augmented feature space using the algorithm mentioned in Section 3.2.3. For notation brevity, we will refer to the former method as "**LB-OR**", and the latter one as "**LB-AR**". In LB-AR, The link matrix in Equation (3.1) takes the form of $\mathbf{R}+\mathbf{R}^\top$, and the weight factor $\lambda = 0.5$.

To compare with the proposed LinkBoost algorithm, two baseline algorithms are implemented. The first baseline algorithm is Feature Representation Augmentation ("**RepAug**" for short). The second one is Co-training ("**Cotrain**" for short). Both baseline algorithm were reviewed in Section 3.2.3.

Since only binary classification model is discussed in this chapter, we tested all the possible class pairs in both datasets. For each class pair, we randomly chose 5% data examples for training and the rest for test. Each classification was repeated 10 times by using different randomly selected training data, and the classification accuracy was averaged over the 10-fold experiments.

3.4.2 Robustness against Sparse Link Information

As discussed in Section 3.4.1, the link information in both Cora and Citeseer datasets is noisy. To further verify the proposed LinkBoost algorithm and its robustness against sparse link information, we gradually decrease the number of links being used, and compare the classification performance of the four algorithms. Table 3.3 and Table 3.4 give the classification accuracies on Cora dataset; Table 3.5 and Table 3.6 give the classification accuracies on Citeseer dataset. In this group of experiments, Support Vector Machine (implemented by SVMLight software) is used as the base supervised classification algorithm to be boosted by the LinkBoost framework.

As we can see, with all the available link information being used, LB-AR and AugRep deliver comparable performance, while overall speaking LB-AR is slightly better. LB-OR is suboptimal among the four methods; and Cotrain almost always performs worst. When the number of links being used is decreasing, the performance of all four methods degrades as expected. However, with the link information getting even sparser gradually, the advantage of LB-AR over AugRep becomes significant, since their performance gap enlarges. Moreover, when only 30% - 10% of links are used, LB-OR also outperforms the AugRep. The above observations suggest that LinkBoost is more robust against the sparseness of link information.

Comparing the performance of LB-OR and LB-AR, i.e., applying LinkBoost on both original feature space or augmented feature space, leads to the following findings: when 80% - 100% links are used, LB-AR performs better than LB-OR; however,

Classes	Model	Percentage of Links Used									
		100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
1 vs 2	AugRep	.900	.898	.891	.888	.883	.875	.866	.859	.849	.841
	Cotrain	.836	.835	.837	.839	.841	.845	.843	.844	.844	.838
	LB-OR	.891	.889	.890	.890	.884	.890	.882	.886	.888	.885
	LB-AR	.911	.911	.905	.907	.900	.896	.897	.889	.889	.898
1 vs 3	AugRep	.912	.910	.907	.902	.897	.893	.883	.873	.863	.846
	Cotrain	.840	.841	.840	.836	.842	.842	.839	.848	.849	.846
	LB-OR	.877	.882	.876	.873	.870	.869	.862	.864	.866	.864
	LB-AR	.911	.908	.908	.903	.895	.891	.880	.874	.869	.857
1 vs 4	AugRep	.841	.835	.826	.822	.810	.798	.794	.779	.759	.748
	Cotrain	.732	.729	.727	.726	.728	.727	.730	.727	.735	.744
	LB-OR	.809	.804	.807	.806	.794	.789	.784	.772	.768	.775
	LB-AR	.850	.840	.836	.832	.815	.809	.802	.804	.779	.782
1 vs 5	AugRep	.847	.844	.841	.832	.826	.816	.807	.790	.777	.770
	Cotrain	.763	.759	.753	.750	.748	.746	.747	.752	.757	.762
	LB-OR	.821	.814	.811	.806	.800	.802	.795	.792	.788	.786
	LB-AR	.846	.840	.845	.833	.824	.814	.807	.789	.784	.782
1 vs 6	AugRep	.944	.941	.935	.929	.921	.909	.897	.876	.862	.861
	Cotrain	.833	.840	.836	.831	.835	.837	.840	.843	.849	.851
	LB-OR	.864	.863	.860	.857	.852	.855	.847	.844	.840	.837
	LB-AR	.945	.939	.932	.923	.920	.904	.891	.874	.857	.853
1 vs 7	AugRep	.894	.889	.885	.877	.868	.857	.843	.828	.817	.803
	Cotrain	.789	.791	.789	.784	.786	.786	.786	.796	.798	.799
	LB-OR	.854	.849	.836	.831	.833	.833	.829	.824	.824	.821
	LB-AR	.899	.896	.890	.884	.876	.868	.854	.842	.841	.830
2 vs 3	AugRep	.889	.879	.861	.851	.832	.802	.783	.745	.703	.705
	Cotrain	.755	.759	.756	.768	.761	.757	.760	.760	.735	.725
	LB-OR	.882	.875	.885	.868	.874	.851	.863	.846	.848	.805
	LB-AR	.891	.883	.869	.860	.844	.833	.830	.827	.815	.806
2 vs 4	AugRep	.852	.853	.847	.839	.830	.819	.801	.788	.776	.760
	Cotrain	.764	.758	.754	.750	.756	.762	.759	.767	.763	.752
	LB-OR	.886	.887	.890	.874	.877	.877	.861	.857	.842	.850
	LB-AR	.942	.937	.934	.938	.920	.926	.902	.884	.893	.879
2 vs 5	AugRep	.741	.732	.729	.725	.721	.710	.713	.706	.700	.687
	Cotrain	.689	.687	.684	.686	.689	.688	.688	.695	.698	.688
	LB-OR	.785	.778	.761	.766	.764	.767	.766	.744	.749	.738
	LB-AR	.808	.815	.807	.799	.775	.774	.756	.742	.733	.737
2 vs 6	AugRep	.924	.920	.905	.889	.883	.862	.841	.825	.803	.771
	Cotrain	.772	.779	.780	.781	.775	.769	.769	.776	.771	.782
	LB-OR	.885	.884	.879	.879	.879	.862	.867	.856	.847	.831
	LB-AR	.934	.919	.919	.888	.896	.896	.883	.875	.859	.843
2 vs 7	AugRep	.732	.733	.731	.723	.717	.703	.690	.679	.670	.654
	Cotrain	.650	.657	.643	.639	.654	.657	.664	.662	.665	.659
	LB-OR	.751	.765	.759	.757	.757	.756	.758	.757	.749	.742
	LB-AR	.773	.773	.767	.760	.760	.763	.734	.728	.747	.752

Table 3.3. Classification Accuracy Comparison on Cora Dataset (Part A).

Classes	Model	Percentage of Links Used									
		100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
3 vs 4	AugRep	.940	.934	.930	.928	.912	.899	.883	.865	.846	.831
	Cotrain	.840	.838	.840	.836	.844	.858	.858	.859	.864	.848
	LB-OR	.902	.904	.895	.902	.906	.908	.897	.904	.899	.889
	LB-AR	.947	.944	.942	.939	.924	.921	.909	.889	.880	.862
3 vs 5	AugRep	.872	.870	.862	.854	.838	.817	.804	.786	.759	.750
	Cotrain	.760	.749	.759	.764	.763	.763	.769	.759	.761	.763
	LB-OR	.831	.832	.826	.828	.812	.807	.806	.799	.778	.781
	LB-AR	.872	.874	.871	.853	.839	.824	.802	.799	.767	.778
3 vs 6	AugRep	.903	.898	.893	.888	.879	.862	.845	.833	.805	.795
	Cotrain	.775	.772	.780	.785	.792	.784	.791	.785	.789	.777
	LB-OR	.856	.847	.844	.847	.845	.839	.839	.830	.829	.815
	LB-AR	.914	.901	.899	.892	.883	.862	.862	.842	.831	.823
3 vs 7	AugRep	.871	.868	.862	.855	.836	.827	.808	.774	.746	.739
	Cotrain	.713	.724	.717	.722	.732	.743	.755	.755	.744	.740
	LB-OR	.797	.782	.783	.790	.784	.787	.806	.788	.800	.789
	LB-AR	.870	.868	.857	.850	.841	.848	.825	.804	.825	.785
4 vs 5	AugRep	.899	.891	.880	.874	.864	.860	.861	.829	.819	.810
	Cotrain	.809	.811	.812	.803	.808	.809	.806	.818	.800	.798
	LB-OR	.844	.800	.818	.819	.827	.831	.831	.838	.841	.831
	LB-AR	.898	.893	.878	.879	.868	.859	.864	.827	.826	.794
4 vs 6	AugRep	.959	.950	.948	.936	.923	.912	.892	.860	.833	.829
	Cotrain	.808	.805	.815	.818	.818	.816	.816	.810	.818	.806
	LB-OR	.923	.892	.898	.910	.915	.909	.914	.915	.919	.923
	LB-AR	.974	.968	.959	.947	.951	.940	.934	.925	.908	.895
4 vs 7	AugRep	.904	.894	.885	.869	.855	.836	.821	.799	.774	.765
	Cotrain	.769	.775	.778	.774	.767	.774	.760	.770	.777	.778
	LB-OR	.829	.806	.799	.809	.813	.819	.810	.832	.832	.844
	LB-AR	.916	.893	.899	.886	.883	.874	.851	.838	.821	.811
5 vs 6	AugRep	.926	.924	.919	.910	.898	.874	.851	.839	.828	.809
	Cotrain	.807	.802	.806	.794	.796	.796	.795	.797	.809	.798
	LB-OR	.836	.822	.826	.835	.840	.847	.847	.853	.862	.866
	LB-AR	.926	.925	.917	.911	.901	.871	.852	.844	.843	.816
5 vs 7	AugRep	.812	.810	.807	.803	.796	.771	.747	.729	.713	.714
	Cotrain	.706	.707	.702	.705	.711	.716	.718	.712	.712	.709
	LB-OR	.782	.739	.745	.764	.749	.760	.769	.759	.776	.774
	LB-AR	.821	.821	.810	.803	.805	.788	.776	.755	.763	.743
6 vs 7	AugRep	.940	.934	.929	.919	.909	.893	.875	.853	.827	.823
	Cotrain	.796	.779	.774	.769	.779	.774	.759	.769	.781	.789
	LB-OR	.846	.828	.833	.837	.831	.839	.843	.850	.843	.843
	LB-AR	.940	.930	.929	.918	.904	.892	.876	.859	.837	.832

Table 3.4. Classification Accuracy Comparison on Cora datasets (Part B).

Classes	Model	Percentage of Links Used									
		100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
1 vs 2	AugRep	.939	.933	.928	.928	.921	.921	.915	.903	.906	.902
	Cotrain	.907	.910	.910	.909	.910	.908	.911	.911	.911	.905
	LB-OR	.938	.934	.933	.931	.929	.923	.927	.929	.927	.923
	LB-AR	.946	.942	.939	.937	.931	.932	.928	.917	.916	.909
1 vs 3	AugRep	.932	.928	.916	.915	.906	.897	.887	.884	.886	.893
	Cotrain	.905	.905	.906	.905	.905	.906	.906	.904	.903	.902
	LB-OR	.921	.923	.918	.918	.916	.917	.911	.913	.906	.905
	LB-AR	.925	.922	.917	.916	.917	.912	.906	.908	.900	.896
1 vs 4	AugRep	.785	.773	.763	.763	.760	.751	.739	.732	.732	.734
	Cotrain	.748	.744	.743	.742	.737	.738	.739	.739	.735	.732
	LB-OR	.760	.759	.756	.755	.750	.755	.751	.753	.755	.755
	LB-AR	.778	.772	.765	.753	.752	.747	.735	.738	.742	.742
1 vs 5	AugRep	.895	.890	.881	.879	.867	.865	.853	.850	.854	.853
	Cotrain	.848	.852	.856	.858	.862	.858	.857	.859	.854	.851
	LB-OR	.859	.860	.863	.863	.860	.870	.870	.865	.856	.853
	LB-AR	.895	.888	.887	.881	.875	.870	.864	.865	.853	.854
1 vs 6	AugRep	.904	.897	.892	.877	.860	.851	.848	.838	.834	.850
	Cotrain	.864	.859	.866	.865	.865	.869	.873	.873	.875	.881
	LB-OR	.883	.860	.863	.863	.860	.870	.870	.865	.856	.853
	LB-AR	.903	.898	.893	.885	.875	.875	.877	.869	.876	.882
2 vs 3	AugRep	.845	.840	.838	.835	.822	.818	.803	.797	.780	.791
	Cotrain	.795	.798	.795	.800	.793	.794	.797	.794	.788	.787
	LB-OR	.807	.803	.810	.811	.803	.804	.794	.794	.797	.794
	LB-AR	.842	.839	.838	.836	.825	.824	.811	.796	.790	.784
2 vs 4	AugRep	.787	.782	.776	.778	.778	.770	.768	.758	.757	.757
	Cotrain	.795	.755	.757	.753	.751	.754	.749	.746	.746	.743
	LB-OR	.780	.773	.777	.773	.773	.773	.773	.773	.775	.776
	LB-AR	.784	.782	.781	.777	.775	.774	.765	.767	.773	.774
2 vs 5	AugRep	.870	.859	.854	.837	.829	.811	.799	.801	.789	.793
	Cotrain	.804	.808	.812	.811	.806	.806	.808	.803	.808	.805
	LB-OR	.841	.846	.843	.843	.845	.838	.830	.824	.817	.804
	LB-AR	.872	.869	.867	.856	.855	.844	.836	.822	.810	.791

Table 3.5. Classification Accuracy Comparison on Citeseer datasets (Part A).

Classes	Model	Percentage of Links Used									
		100%	90%	80%	70%	60%	50%	40%	30%	20%	10%
2 vs 6	AugRep	.834	.831	.828	.824	.815	.793	.769	.760	.742	.765
	Cotrain	.803	.801	.792	.791	.787	.784	.773	.774	.782	.781
	LB-OR	.791	.793	.792	.788	.783	.785	.783	.779	.775	.774
	LB-AR	.829	.829	.824	.819	.807	.782	.778	.765	.748	.754
3 vs 4	AugRep	.766	.765	.759	.757	.756	.756	.753	.751	.750	.749
	Cotrain	.754	.755	.756	.753	.754	.754	.754	.752	.752	.752
	LB-OR	.781	.775	.771	.771	.771	.767	.769	.770	.772	.768
	LB-AR	.770	.773	.766	.761	.761	.759	.755	.752	.758	.767
3 vs 5	AugRep	.890	.883	.869	.856	.840	.830	.823	.820	.817	.821
	Cotrain	.848	.852	.846	.845	.845	.844	.843	.844	.839	.831
	LB-OR	.887	.884	.883	.885	.884	.878	.883	.881	.875	.869
	LB-AR	.905	.899	.890	.885	.881	.873	.869	.866	.862	.863
3 vs 6	AugRep	.874	.865	.854	.844	.839	.831	.821	.815	.803	.809
	Cotrain	.830	.832	.832	.837	.837	.836	.835	.834	.831	.829
	LB-OR	.853	.853	.849	.851	.846	.846	.839	.841	.841	.833
	LB-AR	.878	.870	.861	.854	.854	.844	.842	.845	.831	.835
4 vs 5	AugRep	.726	.726	.721	.713	.709	.703	.705	.709	.704	.701
	Cotrain	.707	.705	.702	.699	.700	.700	.699	.698	.699	.703
	LB-OR	.726	.724	.724	.723	.719	.722	.721	.723	.726	.726
	LB-AR	.731	.724	.725	.725	.726	.721	.726	.721	.717	.719
4 vs 6	AugRep	.713	.712	.711	.711	.710	.708	.710	.709	.709	.710
	Cotrain	.706	.706	.707	.706	.706	.711	.709	.710	.709	.710
	LB-OR	.713	.714	.711	.712	.714	.714	.711	.713	.713	.713
	LB-AR	.714	.711	.711	.710	.706	.705	.707	.707	.709	.712
5 vs 6	AugRep	.876	.861	.851	.848	.836	.820	.809	.805	.807	.814
	Cotrain	.814	.817	.814	.816	.820	.824	.824	.817	.810	.810
	LB-OR	.862	.855	.852	.847	.852	.838	.844	.835	.831	.826
	LB-AR	.882	.864	.859	.862	.854	.848	.839	.835	.820	.822

Table 3.6. Classification Accuracy Comparison on Citeseer datasets (Part B).

when 10% - 30% links are used, LB-OR is slightly better than LB-AR. A possible explanation to these findings is that when significant amount of data examples are involved in links, the classification process is doubly boosted by LB-AR, both from data representation augmentation and the training set augmentation; but when link information are extremely sparse, the representation augmentation in a bag-of-words manner is less reliable, compared to the training set augmention which *selectively* supplements training pool with highly confident predictions. However, how to reliably utilize extremely sparse link information needs further study.

3.4.3 Boosting Power for Supervised Algorithms

To verify the boosting power of the proposed LinkBoost framework, we apply it to several base supervised classifiers, and compare the performance with that of the base classifier itself. In particular, five supervised algorithms are used

- Support Vector Machine ("svm" for short), whose performance was proved to be among the best in many text classification applications.
- J48 decision trees ("j48" for short).
- HyperPipe classifier ("hpp" for short), which constructs for each category a "hypepipe" that contains all points of that category (essentially records the attribute bounds observed for each category). Predictions on test instances are made according to the category that most contains the instance.
- Simple Naive Bayes classifier ("nbs" for short).
- Voted Perceptron ("vp" for short).

For "svm", we used the SVMLight implementation. For the rest four classifiers, we used its implementation in the Weka software[2].

[2] `http://www.cs.waikato.ac.nz/ml/weka/`

Classes	svm	LB-OR	j48	LB-OR	hpp	LB-OR	nbs	LB-OR	vp	LB-OR
1 vs 2	0.843	0.891	0.819	0.930	0.845	0.851	0.853	0.892	0.846	0.874
1 vs 3	0.843	0.877	0.877	0.925	0.813	0.799	0.822	0.849	0.817	0.844
1 vs 4	0.744	0.809	0.733	0.866	0.734	0.733	0.749	0.796	0.712	0.777
1 vs 5	0.772	0.821	0.725	0.834	0.744	0.750	0.767	0.799	0.752	0.789
1 vs 6	0.854	0.864	0.863	0.929	0.769	0.771	0.756	0.820	0.797	0.834
1 vs 7	0.802	0.854	0.768	0.902	0.807	0.821	0.803	0.859	0.805	0.841
2 vs 3	0.732	0.882	0.651	0.915	0.776	0.847	0.772	0.892	0.699	0.812
2 vs 4	0.750	0.886	0.748	0.933	0.800	0.852	0.810	0.897	0.785	0.871
2 vs 5	0.685	0.785	0.680	0.800	0.711	0.733	0.719	0.769	0.676	0.734
2 vs 6	0.773	0.885	0.747	0.913	0.796	0.835	0.791	0.887	0.765	0.858
2 vs 7	0.653	0.751	0.628	0.797	0.712	0.742	0.690	0.767	0.650	0.717
3 vs 4	0.855	0.902	0.837	0.943	0.775	0.789	0.785	0.876	0.791	0.843
3 vs 5	0.758	0.831	0.744	0.892	0.749	0.780	0.736	0.841	0.697	0.757
3 vs 6	0.803	0.856	0.804	0.894	0.732	0.771	0.736	0.834	0.767	0.817
3 vs 7	0.748	0.797	0.741	0.887	0.741	0.758	0.723	0.826	0.730	0.785
4 vs 5	0.809	0.844	0.691	0.888	0.740	0.750	0.750	0.806	0.734	0.792
4 vs 6	0.846	0.923	0.795	0.955	0.807	0.806	0.766	0.895	0.799	0.870
4 vs 7	0.771	0.829	0.681	0.861	0.783	0.790	0.753	0.844	0.742	0.817
5 vs 6	0.806	0.836	0.779	0.916	0.743	0.759	0.743	0.857	0.770	0.823
5 vs 7	0.708	0.782	0.672	0.822	0.723	0.739	0.708	0.804	0.688	0.740
6 vs 7	0.819	0.846	0.724	0.913	0.780	0.783	0.728	0.834	0.787	0.816

Table 3.7. Boosting classification accuracy on Cora datasets.

Classes	svm	LB-OR	j48	LB-OR	hpp	LB-OR	nbs	LB-OR	vp	LB-OR
1 vs 2	0.904	0.938	0.853	0.946	0.785	0.865	0.817	0.903	0.818	0.882
1 vs 3	0.901	0.921	0.870	0.929	0.790	0.838	0.793	0.862	0.832	0.869
1 vs 4	0.741	0.760	0.733	0.772	0.737	0.747	0.738	0.761	0.734	0.755
1 vs 5	0.854	0.859	0.840	0.887	0.749	0.796	0.752	0.815	0.754	0.810
1 vs 6	0.869	0.883	0.842	0.898	0.754	0.803	0.775	0.844	0.782	0.827
2 vs 3	0.786	0.807	0.665	0.824	0.705	0.773	0.723	0.780	0.742	0.773
2 vs 4	0.754	0.780	0.717	0.798	0.777	0.776	0.778	0.821	0.743	0.785
2 vs 5	0.802	0.841	0.667	0.859	0.763	0.809	0.773	0.840	0.726	0.786
2 vs 6	0.785	0.791	0.639	0.800	0.720	0.738	0.728	0.751	0.715	0.741
3 vs 4	0.754	0.781	0.720	0.807	0.768	0.777	0.769	0.789	0.755	0.773
3 vs 5	0.843	0.887	0.717	0.904	0.791	0.842	0.782	0.868	0.771	0.825
3 vs 6	0.819	0.853	0.726	0.864	0.760	0.796	0.762	0.825	0.745	0.797
4 vs 5	0.702	0.726	0.681	0.758	0.724	0.713	0.730	0.759	0.697	0.738
4 vs 6	0.709	0.713	0.656	0.706	0.706	0.704	0.695	0.691	0.663	0.681
5 vs 6	0.813	0.862	0.681	0.872	0.766	0.804	0.774	0.839	0.742	0.797

Table 3.8. Boosting classification accuracy on Citeseer datasets.

The original data represenation is used both in the base classifier and the LinkBoost-boosted classifier.

Table 3.7 and Table 3.8 compare the performance of the base classifiers, and performance of them within the LinkBoost framwork, on all class pairs in Cora and Citeseer datasets respectively. In both tables, each column contains two sub-columns, with the left one under the name of the base classifier giving its classifation accuracy, and the right one under "LB-OR" giving the classification accuracy of the corresponding base classifier within the LinkBoost framework (using original data representations). As we can see, in most cases, LinkBoost framework is able to significantly improve the classification accuracy. These results empirically proves the boosting power of LinkBoost framework, i.e., it is able to improve *any* supervised algorithm by effectively exploiting link information.

3.5 Conclusions

In this chapter, we propose a semi-supervised learning framework, named as "LinkBoost", for boosting classification performance, when side information in the form of links are available. LinkBoost is designed to turn *any* supervised algorithm into a semi-supervised one, and improve its classification performance. Experiments show that LinkBoost is robust against the noisy and sparse nature of link information, and it does improve the classification accuracy of several typical supervised algorithms.

CHAPTER 4

Semi-supervised Clustering with Pairwise Constraints

In this chapter, a novel boosting framework for semi-supervised clustering will be described. Starting with the problem definition of semi-supervised clustering, we will review a few major approaches in previous studies on this problem, then present our boosting idea with formal description of the related algorithms. Experiment results and discussions will be provided, as empirical validation. Finally, we will summarize our work and raise a few issues for future work.

4.1 Problem Definition

Data clustering, also called *unsupervised learning*, is one of the key techniques in data mining that is used to understand and mine the structure of unlabeled data. The idea of improving clustering by side information, sometimes called *semi-supervised clustering* or *constrained data clustering*, has received significant amount of attention in recent studies on data clustering. Often, the side information is presented in the form of pairwise constraints: the *must-link* pairs where data points should belong to the same cluster, and the *mustnot-link* pairs where data points should belong to different clusters.

Table 4.1 summarizes the notations that will be used throughout the chapter.

n	:	Total number of data examples.
d	:	The number of attributes for each data example
$\mathbf{x}_i$	:	A d-dimension vector represents the i-th data example. We also use $\mathbf{x}_i$ to refer to the i-th data example.
$\mathcal{X}$	:	The set of all data examples, i.e. $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{n}$.
$\mathbf{X}$	:	A matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ that gathers the vector representations of all the data examples.
$\mathcal{S}^+$	:	The set of all must-link pairs of data examples.
$\mathbf{S}^+$	:	A $n \times n$ matrix where $S^+_{i,j}$ is one when examples $\mathbf{x}_i$ and $\mathbf{x}_j$ form a must-link pair, and zero otherwise.
$\mathcal{S}^-$	:	The set of all mustnot-link pairs of data examples
$\mathbf{S}^-$	:	An $n \times n$ matrix where $S^-_{i,j}$ is one when examples $\mathbf{x}_i$ and $\mathbf{x}_j$ form a mustnot-link pair, and zero otherwise.
c	:	For classification or clustering problems, c gives the number of classes.
l_i	:	For classification or clustering problems, l_i takes values from the set $\{1, \cdots, c\}$ which indicates the class label for the i-th data example.

Table 4.1. Notations

4.2 Review on Previous Studies

There are two major approaches to semi-supervised clustering: the approach based on constraints satisfaction and the approach based on distance metric learning. The first approach employs the side information to restrict the solution space, and only finds the solution that is consistent with the pairwise constraints. The second approach first learns a distance metric from the given pairwise constraints, and computes the pairwise similarity using the learned distance metric. The computed similarity matrix is then used for data clustering. In this section, we will review some major work in both approaches, recapitulate a few selected representative algorithms, followed by a brief summary on these previous studies.

4.2.1 Approach Based on Constraints Satisfaction

The constraint-based approach for semi-supervised clustering utilizes the side information to restrict the feasible solutions when deciding the cluster assignment. Early work in this category took the side information as the hard constraints, and only considered the cluster assignments that were absolutely consistent with the given pairwise constraints. In [148, 20], the authors proposed the constrained K-means algorithms by adjusting the cluster memberships to be consistent with the pairwise constraints. In [133], a generalized Expectation Maximization (EM) algorithm is proposed to incorporate the pairwise constraints into the EM algorithm. In particular, the cluster assignments that are inconsistent with the constraints are excluded from the partition function when computing the posterior probability for the cluster memberships. One problem with treating the side information as hard constraints is that we may not be able to find feasible solutions that are consistent with all the constraints [44]. To overcome this problem, a number of studies view the side information as *soft* constraints. The key idea is to penalize, not to exclude, the cluster assignments that are inconsistent with the given pairwise constraints. In [15, 108, 16], the authors present probabilistic models for semi-supervised clustering where the pairwise constraints are incorporated into the clustering algorithms through the Bayesian priors. In [102], the authors modified the mixture model for data clustering by redefining the data generation process through the introduction of hidden variables. In [14], the authors extended the framework of semi-supervised clustering by selecting the most informative pairwise constraints to solicit the labeling information. In [45], the authors studied semi-supervised clustering for the hierarchical clustering algorithm.

4.2.2 Approach Based on Distance Metric Learning

Another approach to semi-supervised clustering is to first learn a distance metric from the given pairwise constraints. The pairwise similarity between any two examples is

then computed based on the learned distance metric, and a clustering algorithm is applied to the computed similarity matrix. The key to this approach is to effectively learn a distance metric from the side information. Zhang et al. [165] proposed to learn a distance metric by a linear regression model. Xing et al. [155] formulated the distance metric learning problem as a constrained convex programming problem. This algorithm is extended to the nonlinear case in [98] by the introduction of kernels. Yang et al. [159] proposed a local distance metric algorithm that is designed to address the problem of distance metric learning for multi-modal data distributions. Golderberg et al. [64] presented the neighborhood component analysis algorithm that learns a local distance metric by extending the nearest neighbor classifier. Weinberger [150] presented a large margin nearest-neighbor classifier for distance metric learning that extended the neighborhood component analysis to a maximum margin framework. Discriminative component analysis [74] learned a distance metric by extending the relevance component analysis to effectively explore both the must-link and the mustnot-link constraints simultaneously. In [71, 72], the authors extended the boosting algorithms to learn a distance function from given pairwise constraints. Schultz and Joachims [128] extended the framework of support vector machine to learn distance metrics from the pairwise comparisons.

Finally, a few studies cluster data points by a similarity matrix that is directly modified according to the pairwise constraints. In [96], the authors proposed to modify the similarity matrix by linearly combining the original similarity matrix with the pairwise constraints. Klein et al. [91] proposed to modify the similarity matrix by propagating the pairwise constraints through the nearest neighbors.

4.2.3 Representative Algorithms

We will recapitulate a few representative algorithms proposed for semi-supervised clustering in the following.

CONSTRAINED K-MEANS CLUSTERING ALGORITHM

The constrained K-means algorithm is a modified K-means algorithm, by ensuring *none* of the pairwise constrained is violated during the iterative steps of the K-means algorithm. Specifically, the constrained K-means algorithm can be formulated as follows [148]

Input

- $\mathbf{X}$: matrix for the input data
- $\mathbf{S}^+$: matrix for must-link pairs
- $\mathbf{S}^-$: matrix for mustnot-link pairs

Output: cluster memberships

Algorithm

Step 1 Initialize $C_1, \ldots, C_k$ as the initial cluster centers.

Step 2 For each data point x_i, assign it to the closest cluster center C_j such that

- for all $\mathbf{x}_{i'}$ not belonging to the k-th cluster, $S^+_{i,j} \neq 1$;
- for all $\mathbf{x}_{i'}$ belonging to the k-th cluster, $S^-_{i,j} \neq 1$.

If no C_j satisfies the above rules, fail.

Step 3 Update each cluster center C_j by averaging all the data point d_i in the corresponding cluster.

Step 4 Iterate through Step 2 and Step 3 until convergence.

Step 5 Return cluster memberships.

The main drawback of the above algorithm is that it can fail without yielding a feasible solution. As the algorithm presents, if the cluster assignment (in Step 2) cannot be found to satisfy all the pairwise constraints, the algorithm will stop. Another drawback with the constrained K-means algorithm is that it only makes

efforts to satisfy those "known" constraints, but does not generalize those constraints to the unseen data where the pairwise relationship is "unknown".

Constrained Complete-Link Clustering Algorithm

The data examples involved in pairwise constraints, in general, can be viewed as representative of their local neighborhoods. Having recognized this, it would be natural to try to induce a set of new distance measurements over all the data examples from the limited number of pairwise constraints. This is the basic idea of the work presented in [91], which is also formulated as acquiring prior knowledge "from instance level constraints to space-level constraints". Then a specific clustering algorithm (Complete-Link clustering) is applied with the new distance measurements. The corresponding algorithm is named as Constrained Complete-Link (CCL) algorithm.

In CCL algorithm, the distance measurement between each pair of data examples are generated by explicitly making adjustment on an initial distance matrix computed from the data input patterns. The adjustment is done by first imposing the constraints, and then propagating them to the neighborhood of the constrained examples.

For must-link constraints, imposing the constraints means setting each distance between the must-link pair of data examples to zero. Then, the distance between all other data example pairs are recomputed as the length of shortest path connecting them (allowing using the "zero" length of those must-links). In this way, the must-link gets propagated to their neighborhoods. After these distance adjustment, the triangle inequality still holds, and the resulting distance matrix is still a valid "metric".

For mustnot-links, imposing the constraints means setting each distance between the mustnot-link pair of data examples to a large value. However, further propagating the mustnot-links to their neighborhood while maintaining the adjusted distance matrix a valid "metric" will be computationally expensive. In [91], the propagation

of mustnot-links are not carried out explicitly by adjusting the distance matrix, but claimed to be implicitly done in the merging step of the following Complete-Link clustering procedures [1].

The most notable contribution of the CCL algorithm is its efforts to generalize the instance-level pairwise constraints to space-level distance measurements that can affect data examples beyond the constrained ones. As a results, it is reported in [91] to outperform the constrained K-means algorithm in clustering.

HMRF-KMEANS ALGORITHM

Basu et al. proposed a probabilistic framework based on Hidden Markov Random Fields (HMRFs) that combines constraints satisfaction and distance metric learning [15]. Based on this framework, a partitional clustering algorithm is designed, which is named as HMRF-Kmeans algorithm.

Specifically, the following Hidden Markov Random Field is considered

- A hidden set of random variables $\mathcal{L} = \{l_i\}_{i=1}^n$, where each random variable l_i takes values from the set $\{1, \ldots, c\}$ which is a cluster membership indicator.
- An observable set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, where each random variable $\mathbf{x}_i$ is generated from a conditional probability distribution $\Pr(\mathbf{x}_i|l_i)$.

To incorporate the pairwise constraints, the probability of a particular cluster label configuration is expressed in the following Gibbs distribution form

$$\Pr(\mathcal{L}) = \frac{1}{Z_1} \exp(-\sum_i \sum_j V(i,j)) \tag{4.1}$$

[1] In the Complete-Link algorithm, the distance between two clusters is defined as the maximum distance between data examples from either cluster. Therefore, if $(\mathbf{x}_i, \mathbf{x}_j)$ forms a mustnot-link, merging $\mathbf{x}_k$ with $\mathbf{x}_i$ will result in a mustnot-link practically being constructed between $\mathbf{x}_k$ and $\mathbf{x}_j$.

where Z_1 is a normalizing factor and

$$V(i,j) = \begin{cases} f_+(\mathbf{x}_i, \mathbf{x}_j) & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}^+ \text{ but } l_i \neq l_j \\ f_-(\mathbf{x}_i, \mathbf{x}_j) & \text{if } (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}^- \text{ but } l_i = l_j \\ 0 & \text{otherwise} \end{cases}$$

Here, $f_+(\mathbf{x}_i, \mathbf{x}_j)$ and $f_-(\mathbf{x}_i, \mathbf{x}_j)$ are two non-negative functions that penalize the violations of must-links and mustnot-links, respectively. The intuition behind the above treatment is that higher probability should be assigned to a label configurations that satisfy more pairwise constraints.

The optimal set of cluster labels for all the data example is acquired by solving the following MAP estimation problem

$$\Pr(\mathcal{L}|\mathcal{X}) \propto \Pr(\mathcal{L}) \Pr(\mathcal{X}|\mathcal{L})$$

Since $\Pr(\mathcal{L})$ is given in (4.1), we need to decide $\Pr(\mathcal{X}|\mathcal{L})$ to carry on the MAP estimation. Assuming the set of random variables $\mathcal{X}$ to be conditional independent given the set of hidden variables, i.e.

$$\Pr(\mathcal{X}|\mathcal{L}) = \prod_{\mathbf{x}_i \in \mathcal{X}} \Pr(\mathbf{x}_i|l_i) \tag{4.2}$$

we further parametrize $\Pr(\mathbf{x}_i|l_i)$ as

$$\Pr(\mathbf{x}_i|l_i) \propto \exp(-D(\mathbf{x}_i, \boldsymbol{\mu}_{l_i})) \tag{4.3}$$

where $\boldsymbol{\mu}_k (k = 1, \cdots, c)$ is the cluster representative of the k-th cluster, and $D(\mathbf{x}_i, \boldsymbol{\mu}_{l_i})$ is a distortion measurement between the i-th data example and the l_i-th cluster representative. Such distortion measurement can take various forms, such as cosine similarity or I-divergence, etc..

Combining (4.1) - (4.3), the MAP estimation leads to the following objective function for clustering

$$\max_{\{l_i\}_{i=1}^n, \{\boldsymbol{\mu}_k\}_{k=1}^c} \exp(-\sum_i \sum_j V(i,j)) \exp(-\sum_i D(\mathbf{x}_i, \boldsymbol{\mu}_{l_i}))$$

Note that the above optimization problem is a "incomplete-data problem" since both the set of cluster labels $\{l_i\}_{i=1}^n$ and the cluster representatives $\{\boldsymbol{\mu}_k\}_{k=1}^c$ are unknown in a clustering setting, Expectation Maximization (EM) method is used to find the solution. The EM steps can be formulated into a modified K-means algorithm (see [15] for details).

Metric Pairwise Constrained Kmeans Algorithm

Metric Pairwise Constrained Kmeans (MPCKmeans) is another research attempt which tries to integrate distance metric learning into the iterative steps for clustering [24]. Again, an underlying K-means-style procedure (i.e. iteratively updating cluster memberships and cluster representatives) is assumed for the clustering. The objective function for the related optimization problem is

$$\begin{aligned}\min \quad & \sum_i (\mathbf{x}_i - \boldsymbol{\mu}_{l_i})^\top \mathbf{A}_{l_i} (\mathbf{x}_i - \boldsymbol{\mu}_{l_i}) - \sum_i \log\det(\mathbf{A}_{l_i}) \\ & + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathbf{S}^+} f_+(\mathbf{x}_i,\mathbf{x}_j)\delta(l_i \neq l_j) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathbf{S}^-} f_-(\mathbf{x}_i,\mathbf{x}_j)\delta(l_i = l_j) \quad (4.4)\end{aligned}$$

where $\boldsymbol{\mu}_k (k = 1, \cdots, c)$ is the cluster representative of the k-th cluster, $\mathbf{A}_k$ is a distance matrix for the k-th cluster, and $f_+(\mathbf{x}_i, \mathbf{x}_j)$ and $f_-(\mathbf{x}_i, \mathbf{x}_j)$ are two non-negative functions that penalize the violations of must-links or mustnot-links.

In [24], the two penalty functions are defined in the following forms

$$\begin{aligned} f_+(\mathbf{x}_i,\mathbf{x}_j) &= \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}_{l_i}(\mathbf{x}_i - \mathbf{x}_j) + \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}_{l_j}(\mathbf{x}_i - \mathbf{x}_j) \\ f_-(\mathbf{x}_i,\mathbf{x}_j) &= \frac{1}{2}(\mathbf{x}'_{l_i} - \mathbf{x}''_{l_i})^\top \mathbf{A}_{l_i}(\mathbf{x}'_{l_i} - \mathbf{x}''_{l_i}) - \frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}_{l_j}(\mathbf{x}_i - \mathbf{x}_j) \end{aligned}$$

where $(\mathbf{x}'_{l_i}, \mathbf{x}''_{l_i})$ is the maximally separated pair of the points according to the l_i-th distance matrix $\mathbf{A}_{l_i}$ (note that we only care about f_- when $l_i = l_j$). The function definition of f_+ allow a larger penalty imposed on violating a must-link constraints between a pair of distant data examples than that between a pair of close data examples. This reflects the intuition that if two must-link examples are measured as

far from each other by a distance metric (i.e. a large value of $(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}_{l_i}(\mathbf{x}_i - \mathbf{x}_j)$ or $(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}_{l_j}(\mathbf{x}_i - \mathbf{x}_j)$), we want to put more penalty for this situation so that the corresponding distance metric can to be adjusted dramatically. A similar argument can be found for the function definition of f_-.

Note that there are four terms in the objective function (4.4). The first term addresses the expectation on the compactness of the data clusters, as in the K-means algorithm, but allowing each cluster to take different shapes; the second term regularizes the learned distance matrices; the third and the fourth term try to enforce the pairwise constraints.

Similar to the HMRF-Kmeans algorithm, EM algorithm is used to solve the optimization problem (4.4), which can be formulated into a modified Kmeans algorithm (see [24] for details).

The main advantages of MPCKmeans algorithm are: 1) instead of using a fixed distance metric for clustering, it allows new distance metrics being learned during the clustering procedures and hence improves clustering performance; 2) it allows different distance metrics being learned for different clusters, so that clusters in different shapes are possible to be detected. As reported in [24], MPCKmeans outperforms several purely semi-supervised distance metric learning methods and purely semi-supervised clustering methods in data clustering applications.

4.2.4 Summary

Although a large number of studies have been devoted to semi-supervised clustering, most of them focus on designing special clustering algorithms that can effectively exploit the pairwise constraints. For instance, algorithms in [15, 108, 16] are designed to improve the probabilistic models for data clustering by incorporating the pairwise constraints into the priors of the probabilistic models; the constrained K-means algorithm [148] exploits the pairwise constraints by adjusting the cluster memberships to

be consistent with the given constraints. However, it is often the case that we have a specific clustering algorithm that is specially designed for the target domain, and we are interested in improving the accuracy of this clustering algorithm by the available side information. This motivates us to design a meta-algorithm that is able to improve *any* given clustering algorithm by the pairwise constraints. It is important to note that such a meta-algorithm should not make any assumption about the underlying clustering algorithm, so that it can be applicable to any clustering algorithm. To this end, we propose a boosting framework for data clustering. More details will be given in the following section.

4.3 Boosting Clustering

With pairwise constraints available, it is always desirable if we can utilize such side information to improve clustering performance, no matter which clustering algorithm is used. To this end, we propose a general boosting framework, termed as **BoostCluster**. In this section, we will first illustrate the main idea of boosting clustering, followed by a formal description on the framework, including the related optimization problem and solution, two variations of the meta-algorithm for boosting, and related discussions on the scalability issue.

Let $\mathcal{A}$ denote the given clustering algorithm to be improved. In order to make this framework general, we treat the clustering algorithm $\mathcal{A}$ as a black box that only takes the data representation of all examples as its input and outputs the cluster memberships for the given examples. Note in this work, we assume that the number of clusters is given.

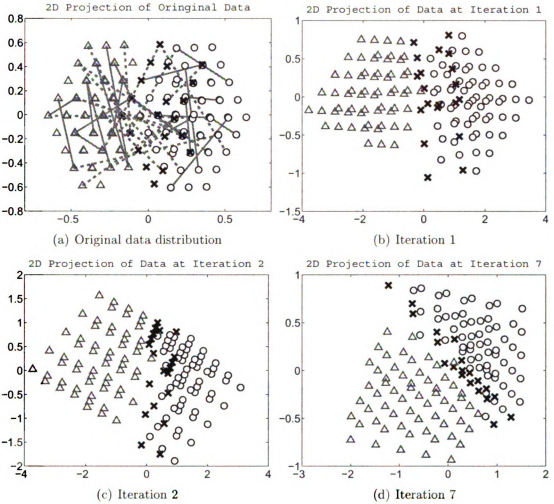
4.3.1 Main Idea

Although a number of boosting algorithms (e.g., [55]) have been successfully applied to supervised learning, extending boosting algorithms to data clustering is signifi-

cantly more challenging. The key difficulty is how to influence an arbitrary clustering algorithm with the given pairwise constraints. To overcome this challenge, we propose to encode the side information into the data representation that is the input to the clustering algorithm. More specifically, we will first find the subspace in which data points of the must-link pairs are close to each other while data points of the mustnot-link pairs are far apart. A new data representation is then generated by projecting all the data points into the subspace, that is used by the given clustering algorithm to find the appropriate cluster assignments. Furthermore, the procedure for identifying the appropriate subspace based on the remaining unsatisfied constraints, and the procedure for clustering data points using the newly generated data representation will alternate iteratively till most of the constraints are satisfied.

Figure 4.1 illustrates the idea of iterative data projection. The data points used in this illustration are sampled from the "scale" dataset that will be described later in Section 4.4.1. They belong to three clusters that are labeled in Figure 4.1 by legends $\triangle$, $\circ$, and $\times$, respectively. A partitional clustering algorithm is used in this illustration. Sub-figure (a) shows the original data distribution projected into a 2D space that is generated by Principle Component Analysis (PCA). We clearly see that many data points of the cluster $\times$ overlap heavily with the data points of the clusters $\triangle$ and $\circ$, and they are difficult to be well separated. The must-link and mustnot-link constraints are indicated in sub-figure (a) by solid lines and dotted lines, respectively. Sub-figures (b)-(d) illustrate the projected data distributions based on the new representations that are generated by the proposed boosting framework in iteration 1, 2, and 7, respectively. Evidently, the data representations generated in different iterations are helpful in separating the data points in the cluster $\times$ from those in the other two clusters.

In practice, the whole boosting idea will work in the following way: the original data representation and the pairwise constraints will be used as input in the proposed



(a) Original data distribution (b) Iteration 1

(c) Iteration 2 (d) Iteration 7

Figure 4.1. An example illustrating the idea of iterative data projections. Sub-figure (a) shows the original data distribution, projected to the space spanned by its two principal components; Sub-figures (b)-(d) show the data distributions based on the new representations in the projected subspaces that are generated in iteration 1, 2, and 7.

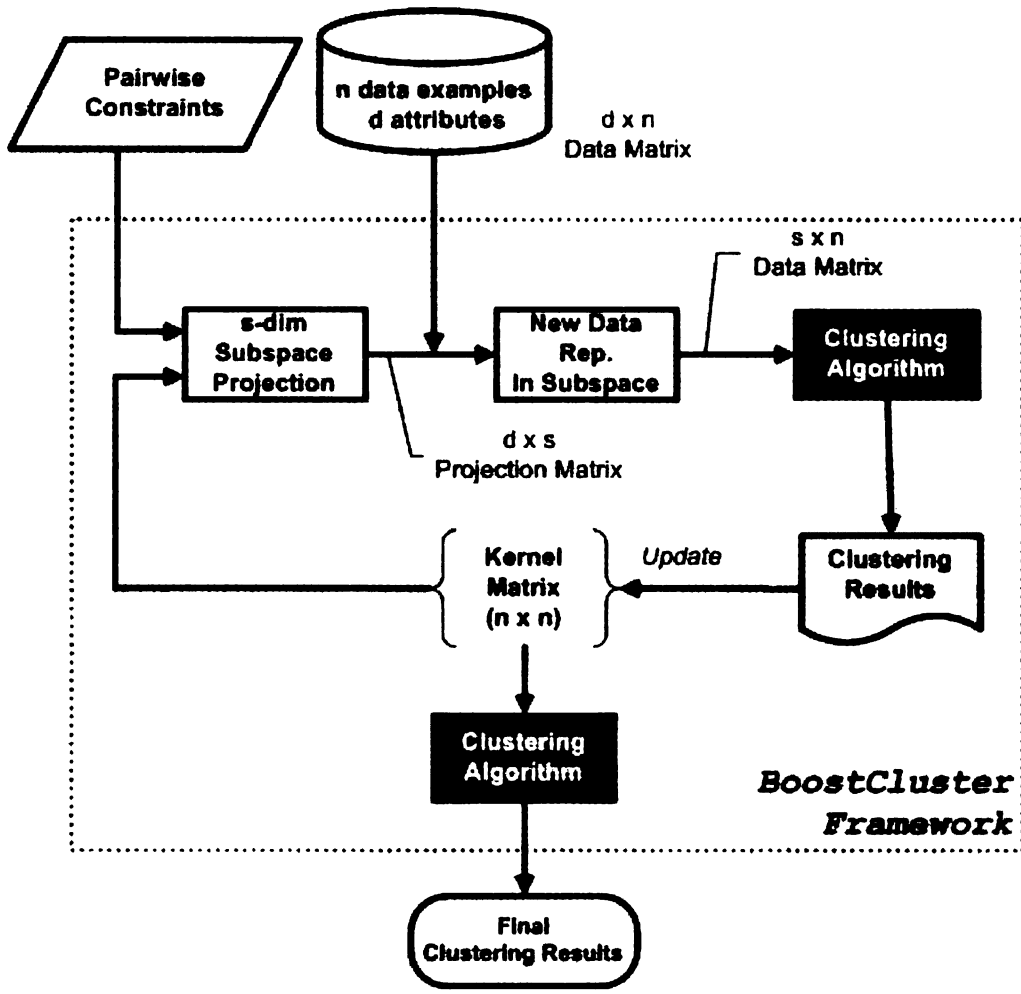


Figure 4.2. The flowchart of the BoostCluster framework.

boosting framework; then in a iterative manner, new data representations will be generated and be fed into the "black-box" clustering algorithm, whose results will in turn be used to find a subspace where data representations can be generated for the next iteration; during the iterative process, a kernel similarity matrix gets updated. The kernel similarity matrix, which can be seen as learned from the above boosting procedure, incorporates the side information gathered from the pairwise constraints and will improve the clustering performance. Figure 4.2 presents a flowchart that illustrates the working mechanism of the boosting framework. More details will be provided in the following subsections.

4.3.2 Objective Function

The first step in designing boosting algorithm is to construct an appropriate objective function. Note that, as described in the introduction section, the goal of the boosting algorithm is to identify the subspace that keeps the data points in the must-link pairs close to each other, and keeps the data points from the mustnot-link pairs well separated. To this end, we introduce the kernel similarity matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$, where $K_{i,j} \geq 0$ indicates the confidence of assigning examples $\mathbf{x}_i$ and $\mathbf{x}_j$ to the same cluster. Then, our goal is to iteratively construct this kernel similarity matrix by using the clustering algorithm $\mathcal{A}$ and the pairwise constraints in $\mathbf{S}^+$ and $\mathbf{S}^-$.

Since the ideal kernel matrix $\mathbf{K}$ is expected to assign a large value to examples in a must-link pair and a small value to examples in a mustnot-link pair, we propose to minimize the following objective function:

$$\mathcal{L}^{BCP} = \sum_{i,j=1}^{n} \sum_{a,b=1}^{n} S^+_{i,j} S^-_{a,b} \exp(K_{a,b} - K_{i,j}) \tag{4.5}$$

In the above, each term within the summation compares $K_{a,b}$, i.e., the similarity between two points from an mustnot-link pair, to $K_{i,j}$, i.e., the similarity between two data points from a must-link pair. By minimizing the objective function in (4.5),

we will ensure that all the data points in the must-link pairs are more similar to each other than the data points in the mustnot-link pairs.

The objective function in (4.5) can also be written as:

$$\mathcal{L}^{BCP} = \left(\sum_{i,j=1}^{n} S_{i,j}^{+} \exp(-K_{i,j}) \right) \left(\sum_{a,b=1}^{n} S_{a,b}^{-} \exp(K_{a,b}) \right) \tag{4.6}$$

The above objective function is a product of two terms: the first term, i.e., $\sum_{i,j=1}^{n} S_{i,j}^{+} \exp(-K_{i,j})$, measures the inconsistency between the kernel similarity matrix K and the must-link constraints; the second term, i.e., $\sum_{a,b=1}^{n} S_{a,b}^{-} \exp(K_{a,b})$, measures the inconsistency between the kernel similarity matrix K and the mustnot-link constraints. Thus, by minimizing the product of the two terms, we enforce the kernel matrix K to be consistent with the given pairwise constraints.

Instead of multiplying the two inconsistency measurements as in (4.6), we can also define the objective function to be the sum of the two terms, i.e.,:

$$\mathcal{L}^{BCS} = \sum_{i,j=1}^{n} S_{i,j}^{+} \exp(-K_{i,j}) + c \sum_{a,b=1}^{n} S_{a,b}^{-} \exp(K_{a,b}) \tag{4.7}$$

The parameter c in (4.7) balances the two inconsistency measurements. To differentiate these two objective functions, we refer to the objective function in (4.6) as *BoostCluster by product*, or **BCP** for short, and the objective function in (4.7) as *BoostCluster by sum*, or **BCS** for short.

4.3.3 The BoostCluster Framework

We will first describe the efficient optimization algorithm for BCP, followed by the algorithm for BCS.

The BCP Algorithm

To boost the performance of a clustering algorithm $\mathcal{A}$ given a set of pairwise constraints, we follow the idea of boosting algorithms by iteratively improving the kernel similarity matrix $\mathbf{K}$. Let $\mathbf{K}$ denote the current kernel similarity matrix. Let

$\mathbf{\Delta} \in \{0,1\}^{n \times n}$ denote the incremental kernel similarity matrix that is inferred from the clustering results generated by the algorithm $\mathcal{A}$. In particular, $\Delta_{i,j} = 1$ when both $\mathbf{x}_i$ and $\mathbf{x}_j$ are assigned to the same cluster and $\Delta_{i,j} = 0$ otherwise. The overall kernel matrix $\mathbf{K}'$ is a linear combination of $\mathbf{K}$ and $\mathbf{\Delta}$, i.e.,

$$\mathbf{K}' = \mathbf{K} + \alpha\mathbf{\Delta} \tag{4.8}$$

where $\alpha \geq 0$ is the combination weight. Then, the objective function $\mathcal{L}^{BCP}$ for the combined kernel $\mathbf{K}'$, denoted by $\mathcal{L}^{BCP}(\mathbf{K}')$, is written as:

$$\begin{aligned} \mathcal{L}^{BCP}(\mathbf{K}') &= \sum_{i,j=1}^{n} \sum_{a,b=1}^{n} S^{+}_{i,j} S^{-}_{a,b} \exp(K'_{a,b} - K'_{i,j}) \\ &= \sum_{i,j=1}^{n} \sum_{a,b=1}^{n} p_{i,j} q_{a,b} \exp(-\alpha(\Delta_{i,j} - \Delta_{a,b})) \end{aligned} \tag{4.9}$$

where

$$p_{i,j} = S^{+}_{i,j} \exp(-K_{i,j}) \tag{4.10}$$

$$q_{a,b} = S^{-}_{a,b} \exp(K_{a,b}) \tag{4.11}$$

In the above, $p_{i,j}$ measures the inconsistency between the kernel matrix $\mathbf{K}$ and the must-link pair $(\mathbf{x}_i, \mathbf{x}_j)$, and $q_{a,b}$ measures the inconsistency between $\mathbf{K}$ and the mustnot-link pair $(\mathbf{x}_a, \mathbf{x}_b)$.

We then employ the Jensen's inequality to obtain an upper bound for the function in (4.9), i.e.,

$$\exp\left(\sum_{i=1}^{m} p_i x_i\right) \leq \sum_{i=1}^{m} p_i \exp(x_i)$$

where $p_i \geq 0, i = 1, 2, \ldots, m$ and $\sum_{i=1}^{m} p_i = 1$. Using the above inequality, an upper bound for $\exp(\Delta_{a,b} - \Delta_{i,j})$ can be constructed as follows

$$\begin{aligned} &\exp(-\alpha(\Delta_{i,j} - \Delta_{a,b})) \\ &= \exp\left(-3\alpha \frac{\Delta_{i,j} - \Delta_{a,b} + 1}{3} + 3\alpha\frac{1}{3} + 0 \times \frac{\Delta_{a,b} - \Delta_{i,j} + 1}{3}\right) \\ &\leq \frac{\Delta_{i,j} - \Delta_{a,b} + 1}{3} \exp(-3\alpha) + \frac{1}{3}\exp(3\alpha) + \frac{\Delta_{a,b} - \Delta_{i,j} + 1}{3} \end{aligned} \tag{4.12}$$

In the first step of the above derivation, we rewrite $\alpha(\Delta_{a,b} - \Delta_{i,j})$ as a summation over the probability distribution of $((\Delta_{a,b} - \Delta_{i,j} + 1)/3, (\Delta_{i,j} - \Delta_{a,b} + 1)/3, 1/3)$. Note that $(\Delta_{a,b} - \Delta_{i,j} + 1) \geq 0$ since $0 \leq \Delta_{i,j} \leq 1$. Using the upper bound in (4.12), we can now bound the objective function of BCP in (4.9) as follows

$$\begin{aligned}
\mathcal{L}^{BCP}(\mathbf{K}') &= \sum_{i,j=1}^{n}\sum_{a,b=1}^{n} p_{i,j}q_{a,b}\exp(-\alpha(\Delta_{i,j} - \Delta_{a,b})) \\
&\leq \frac{\exp(-3\alpha) - 1}{3}\sum_{i,j=1}^{n}\sum_{a,b=1}^{n} p_{i,j}q_{a,b}(\Delta_{i,j} - \Delta_{a,b}) \\
&\quad + \frac{\exp(3\alpha) + \exp(-3\alpha) + 1}{3}\sum_{i,j=1}^{n}\sum_{a,b}^{n} p_{i,j}q_{a,b} \\
&= \frac{\exp(3\alpha) - 1}{3}\sum_{i,j=1}^{n}\Delta_{i,j}\left(p_{i,j}\sum_{a,b=1}^{n} q_{a,b} - q_{i,j}\sum_{a,b=1}^{n} p_{a,b}\right) \\
&\quad + \frac{\exp(3\alpha) + \exp(-3\alpha) + 1}{3}\sum_{i,j=1}^{n}\sum_{a,b}^{n} p_{i,j}q_{a,b} \qquad (4.13)
\end{aligned}$$

We can simplify the above expression by defining a matrix $\mathbf{T}$ as follows

$$T_{i,j} = \frac{p_{i,j}}{\sum_{a,b=1}^{n} p_{a,b}} - \frac{q_{i,j}}{\sum_{a,b=1}^{n} q_{a,b}}$$

The elements in matrix $\mathbf{T}$ measure the inconsistency between kernel matrix $\mathbf{K}$ and the pairwise constraints: a large positive value for $T_{i,j}$ indicates that $\mathbf{K}$ is inconsistent with the must-link pair $(\mathbf{x}_i, \mathbf{x}_j)$; similarly, a large negative value for $T_{a,b}$ indicates that $\mathbf{K}$ is inconsistent with the mustnot-link pair $(\mathbf{x}_a, \mathbf{x}_b)$. Using the notation of matrix $\mathbf{T}$, the upper bound for $\mathcal{L}^{BCP}$ in (4.13) becomes

$$\mathcal{L}^{BCP}(\mathbf{K}') \leq \mathcal{L}^{BCP}(\mathbf{K}) \times \left\{\frac{[\exp(3\alpha) + \exp(-3\alpha) + 1]}{3} - \frac{[1 - \exp(-3\alpha)]\mathrm{tr}(\mathbf{T}\boldsymbol{\Delta}^{\top})}{3}\right\} \qquad (4.14)$$

where

$$\mathcal{L}^{BCP}(\mathbf{K}) = \sum_{i,j=1}^{n} \sum_{a,b=1}^{n} p_{i,j} q_{a,b}$$

$$\text{tr}(\mathbf{T}\boldsymbol{\Delta}^\top) = \sum_{i,j=1}^{n} T_{i,j} \Delta_{i,j}$$

Note that when $\alpha = 0$, the right side of (4.14) becomes $\mathcal{L}^{BCP}(\mathbf{K})$, i.e., the objective function of the previous iteration. Thus, by minimizing the upper bound in (4.14) with respect to α, we are guaranteed to have $\mathcal{L}^{BCP}(\mathbf{K}') \leq \mathcal{L}^{BCP}(\mathbf{K})$, thus reducing the objective function at successive iterations.

As suggested by the inequality in (4.14), to effectively reduce the objective function $\mathcal{L}^{BCP}$, we need to maximize the term $\text{tr}(\mathbf{T}\boldsymbol{\Delta}^\top)$. We further assume that the incremental kernel matrix $\boldsymbol{\Delta}$ can be approximated by a linear projection of the input data $\mathbf{X}$, i.e.,

$$\boldsymbol{\Delta} \approx (\mathbf{P}^\top \mathbf{X})^\top (\mathbf{P}^\top X) = \mathbf{X}^\top \mathbf{P}\mathbf{P}^\top \mathbf{X}$$

where $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_s)$ is the projection matrix ($s \leq d$) with each $\mathbf{p}_i \in \mathbb{R}^d$ specifying a different projection direction. Using the above expression, $\text{tr}(\mathbf{T}\boldsymbol{\Delta}^\top)$ can be written as

$$\text{tr}(\mathbf{T}\boldsymbol{\Delta}^\top) \approx \text{tr}(\mathbf{P}^\top \mathbf{X}\mathbf{T}\mathbf{X}^\top \mathbf{P}) \tag{4.15}$$

If we enforce orthogonality between any two projection vectors, i.e., $\mathbf{p}_i^\top \mathbf{p}_j = \delta(i,j)$, the optimal solution for $\mathbf{p}_i$ that maximizes the expression in (4.15) is the i-th maximum eigenvector of matrix $\mathbf{X}\mathbf{T}\mathbf{X}^\top$. Let $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^{s}$ denote the top s principal eigenvalues and eigenvectors of matrix $\mathbf{X}\mathbf{T}\mathbf{X}^\top$. Then, the optimal projection matrix $\mathbf{P}$ is constructed as

$$\mathbf{P} = (\sqrt{\lambda_1}\mathbf{v}_1, \sqrt{\lambda_2}\mathbf{v}_2, \ldots, \sqrt{\lambda_s}\mathbf{v}_s) \tag{4.16}$$

Using the projection computed in (4.16), we generate a new data representation as $\mathbf{X}' = \mathbf{P}^\top \mathbf{X}$. This new representation $\mathbf{X}'$ will be input to the clustering algorithm

Input

- $\mathbf{X}$: matrix for the input data
- $\mathcal{A}$: the given clustering algorithm
- s: the number of principal eigenvectors used for projection
- $\mathbf{S}^+$: matrix for must-link pairs
- $\mathbf{S}^-$: matrix for mustnot-link pairs

Output: cluster memberships

Algorithm

- Initialize $K_{i,j} = 0$ for any $i, j = 1, 2, \ldots, n$.
- For $t = 1, 2, \ldots, T$
  - Compute $p_{i,j}$ and $q_{i,j}$ using (4.10) and (4.11).
  - Compute matrix $\mathbf{T}$ using (4.15) for BCP and using (4.23) for BCS.
  - Compute the top s eigenvectors and eigenvalues $\{(\lambda_i, \mathbf{v}_i)\}_{i=1}^{s}$ of $\mathbf{T}$.
  - Construct the projection matrix $\mathbf{P}$ using (4.16), and generate the new data representation $\mathbf{X}'$ by projecting the input data $\mathbf{X}$ onto $\mathbf{P}$.
  - Run the clustering algorithm $\mathcal{A}$ using the new data representation $\mathbf{X}'$. Compute the matrix $\mathbf{\Delta}$ with $\Delta_{i,j} = 1$ when $\mathbf{x}_i$ and $\mathbf{x}_j$ are grouped into the same cluster by $\mathcal{A}$, and zero otherwise.
  - Compute α using (4.18) for BCP and using (4.24) for BCS.
  - Update the kernel similarity matrix $\mathbf{K}$ as $\mathbf{K} + \alpha\mathbf{\Delta} \rightarrow \mathbf{K}$
- Run the clustering algorithm $\mathcal{A}$ with the kernel matrix $\mathbf{K}$ (if $\mathcal{A}$ does not take a kernel similarity matrix as input, a data representation can be generated by the first $s + 1$ eigenvectors of the matrix $\mathbf{K}$).

Figure 4.3. Boosting algorithm for BCP and BCS

$\mathcal{A}$ to generate new cluster memberships. The resulting cluster memberships are then used to compute the incremental kernel matrix $\mathbf{\Delta}$.

In addition to the projection matrix $\mathbf{P}$, another important question is how to compute the optimal α. We can estimate the optimal α by minimizing the upper bound in (4.14), which leads to $\alpha = \log[1 + \mathrm{tr}(\mathbf{T}\mathbf{\Delta}^\top)]/6$. However, we can further improve the estimation of α by minimizing the original objective function in (4.6),

which is

$$\begin{aligned}
\mathcal{L}^{BCP}(\mathbf{K}') &= \left(\sum_{i,j=1}^{n} S_{i,j}^{+}\exp(-K_{i,j})\right)\left(\sum_{i,j=1}^{n} S_{i,j}^{-}\exp(K_{i,j})\right) \\
&= \left(\sum_{i,j=1}^{n} p_{i,j}^{+}\exp(-\alpha\Delta_{i,j})\right)\left(\sum_{i,j=1}^{n} q_{i,j}^{-}\exp(\alpha\Delta_{i,j})\right) \\
&= \left(\sum_{i,j=1}^{n} p_{i,j}\delta(\Delta_{i,j},0) + \sum_{i,j=1}^{n} p_{i,j}\delta(\Delta_{i,j},1)\exp(-\alpha)\right) \\
&\quad \times \left(\sum_{i,j=1}^{n} q_{i,j}\delta(\Delta_{i,j},0) + \sum_{i,j=1}^{n} q_{i,j}\delta(\Delta_{i,j},1)\exp(\alpha)\right) \quad (4.17)
\end{aligned}$$

It is not difficult to show that the optimal α that maximizes the above expression is:

$$\alpha = \frac{1}{2}\log\left(\frac{\sum_{i,j=1}^{n} p_{i,j}\delta(\Delta_{i,j},1)}{\sum_{i,j=1}^{n} p_{i,j}\delta(\Delta_{i,j},0)} \times \frac{\sum_{i,j=1}^{n} q_{i,j}\delta(\Delta_{i,j},0)}{\sum_{i,j=1}^{n} q_{i,j}\delta(\Delta_{i,j},1)}\right) \quad (4.18)$$

Figure 4.3 summarizes the proposed BCP (and BCS) algorithm.

Discussions on Efficiency and Scalability

Similar to most boosting algorithms, we can show that the objective function of the proposed BCP algorithm is reduced exponentially, as shown by the following theorem.

Theorem 1 *Let $\mathbf{\Delta}^1, \mathbf{\Delta}^2, \ldots, \mathbf{\Delta}^T$ be the incremental kernel matrices computed from the clustering results by running the boosting algorithm (in Figure 4.3). Then, the objective function after T iterations, i.e., $\mathcal{L}_T^{BCP}$, is bounded as follows:*

$$\mathcal{L}_T^{BCP} \leq \left(\sum_{i,j=1}^{n} S_{i,j}^{+}\right)\left(\sum_{i,j=1}^{n} S_{i,j}^{-}\right)\prod_{t=1}^{T}(1-\gamma_t), \quad (4.19)$$

where

$$\gamma_t = \frac{(\sqrt{A_t D_t} - \sqrt{B_t C_t})^2}{(A_t + B_t)(C_t + D_t)}$$

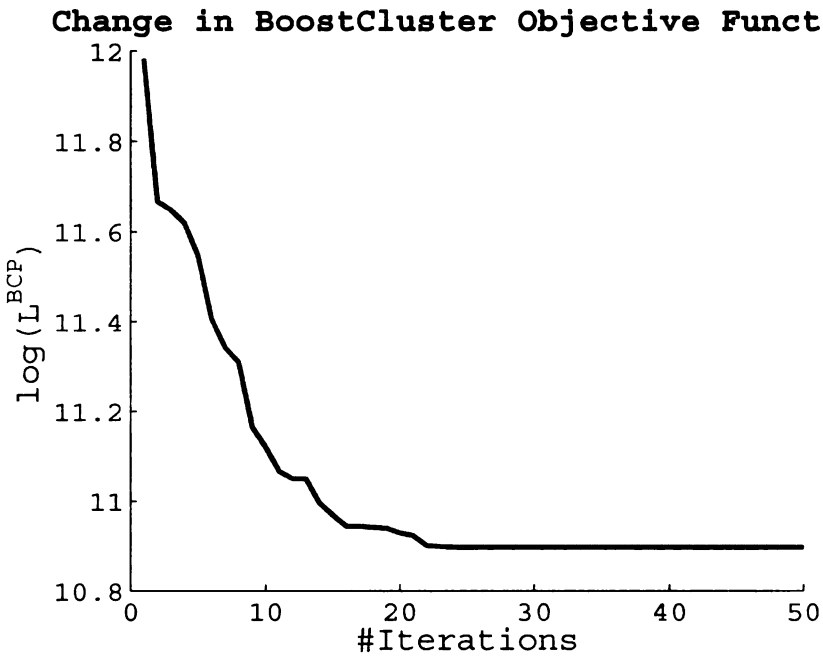


Figure 4.4. An example of BCP objective function vs. number of iterations.

A_t, B_t, C_t, and D_t are non-negative constants, and are computed as

$$A_t = \sum_{i,j=1}^{n} p_{i,j}^t \delta(\Delta_{i,j}^t, 0),\ B_t = \sum_{i,j=1}^{n} p_{i,j}^t \delta(\Delta_{i,j}^t, 1)$$
$$C_t = \sum_{i,j=1}^{n} q_{i,j}^t \delta(\Delta_{i,j}^t, 0),\ D_t = \sum_{i,j=1}^{n} q_{i,j}^t \delta(\Delta_{i,j}^t, 1)$$

where $p_{i,j}^t$ and $q_{i,j}^t$ are computed according to (4.10) and (4.11) using the kernel matrix $\mathbf{K}$ at the t-th iteration.

The above theorem can be proved directly by using the expression in (4.17) and the expression for α in (4.18). Figure 4.4 shows the change in the BCP algorithm's objective function observed in one of our experiments. As can be seen, the objective function converges very fast, and becomes very flat after around 22 iterations. In our experiments, our BCP algorithm usually converges within 25 iterations.

In terms of analyzing the scalability of the BCP algorithm, it is easy to find that the most crucial step is finding the projection matrix $\mathbf{P}$ since it seems to involve a

huge amount of computation cost. As in (4.15), we need to compute $\mathbf{XTX}^\top$, which seems to be computationally expensive, especially when the number of examples (i.e., n) is large. However, it is important $\mathbf{XTX}^\top$ only involves the examples used in the pairwise constraints. This is because $\mathbf{XTX}^\top$ can also be written as:

$$\mathbf{XTX}^\top = \sum_{i,j=1}^{n} T_{i,j}\mathbf{x}_i\mathbf{x}_j^\top \tag{4.20}$$

Since $T_{i,j}$ is nonzero only when the example pair $(\mathbf{x}_i, \mathbf{x}_j)$ are used by the constraint, the above calculation only involves a very small portion of the entire example pairs. Thus, $\mathbf{XTX}^\top$ can be computed efficiently as long as the number of labeled pairs is relatively small.

After $\mathbf{XTX}^\top$ is computed, the computational cost in constructing the projection matrix P mainly arises from computing the principal eigenvectors and eigenvalues of $\mathbf{XTX}^\top$, particularly when the dimensionality of the feature space is high. For instance, for text categorization, each document is represented by a vector of over $100,000$ word features, and the size of matrix $\mathbf{XTX}^\top$ is over $100,000 \times 100,000$. A straightforward approach is to reduce the dimensionality before running the proposed algorithm. However, most dimensionality reduction algorithms that are capable of handling high dimensional space are unsupervised, and therefore are unable to exploit the pairwise constraints. Here, we propose an algorithm that is able to efficiently compute the eigenvectors of $\mathbf{XTX}^\top$ when the input dimensionality is high. We first realize that each eigenvector of $\mathbf{XTX}^\top$ has to lie in the space that is spanned by the examples used by the constraints. More specifically, we denote by $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_m)$ the subset of m examples that are involved in the constraints. Then, the eigenvector $\mathbf{v}_i$ can be written as a linear combination of $\{\tilde{\mathbf{x}}_i\}_{i=1}^m$, i.e.,

$$\mathbf{v}_i = \sum_{k=1}^{m} w_{i,k}\tilde{\mathbf{x}}_k = \tilde{\mathbf{X}}\mathbf{w}_i \tag{4.21}$$

More generally, we have

$$\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2 \ldots, \mathbf{v}_s) = \tilde{\mathbf{X}}(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_s) = \tilde{\mathbf{X}}\mathbf{W}.$$

The proof of this result can be found in Appendix A.1. We furthermore denote by $\tilde{\mathbf{T}}$ the pairwise constraints, where $\tilde{T}_{i,j}$ gives the pairwise constraint between $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$. Then, $\mathbf{w}_i, i = 1, 2, \ldots, s$ correspond to the first s principal eigenvectors of the following generalized eigenvector problem

$$\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \tilde{\mathbf{T}} \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \mathbf{w}_i = \lambda_i \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \mathbf{w}_i \tag{4.22}$$

Note that since $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \tilde{\mathbf{T}} \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$ and $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$ are matrices of $m \times m$, the cost of computing the eigenvectors is independent of the dimensionality of the input space. The proof of the above result can be found in Appendix A.2.

The BCS Algorithm

The derivation of the BCS algorithm is very similar to the BCP algorithm. First, we compute the BCS objective function for $K' = K + \alpha\Delta$ as follows:

$$\begin{aligned} \mathcal{L}^{BCS}(\mathbf{K}') &= \sum_{i,j=1}^{n} S^+_{i,j} \exp(-K'_{i,j}) + cS^-_{i,j} \exp(K_{i,j}) \\ &= \sum_{i,j=1}^{n} p_{i,j} \exp(-\alpha\Delta_{i,j}) + cq_{i,j} \exp(\alpha\Delta_{i,j}) \end{aligned}$$

Using the Jensen's inequality, we have

$$\begin{aligned} \exp(-\alpha\Delta_{i,j}) &= \exp\left(-3\alpha\frac{\Delta_{i,j}+1}{3} + 3\alpha\frac{1}{3} + 0\frac{1-\Delta_{i,j}}{3}\right) \\ &\leq \frac{\Delta_{i,j}+1}{3}\exp(-3\alpha) + \frac{1}{3}\exp(3\alpha) + \frac{1-\Delta_{i,j}}{3} \\ &= \frac{1+\exp(3\alpha)+\exp(-3\alpha)}{3} - \Delta_{i,j}\frac{1-\exp(-3\alpha)}{3} \end{aligned}$$

Similarly, the upper bound for $\exp(\alpha\Delta_{i,j})$ is the following expression

$$\exp(\alpha\Delta_{i,j}) \leq \frac{1+\exp(3\alpha)+\exp(-3\alpha)}{3} + \Delta_{i,j}\frac{1-\exp(-3\alpha)}{3}$$

Using the above inequalities, the BCS objective function $\mathcal{L}^{BCS}(K')$ is upper bounded by the following expression:

$$\mathcal{L}^{BCS}(\mathbf{K}') \leq \frac{1+\exp(3\alpha)+\exp(-3\alpha)}{3}\mathcal{L}^{BCS}(K) - \frac{1-\exp(-3\alpha)}{3}\mathrm{tr}(T_s\Delta^\top)$$

where $\mathbf{T}_s$ is defined

$$[\mathbf{T}_s]_{i,j} = p_{i,j} - cq_{i,j} \tag{4.23}$$

Note that the above definition is similar to $\mathbf{T}$ in (4.15). The key difference is that $p_{i,j}$ and $q_{i,j}$ are normalized in (4.15) while they are not in the above expression. Similar to the BCP algorithm, we compute the top s principal eigenvectors of T_s to construct the projection matrix, and the projected input data is sent to the algorithm $\mathcal{A}$ for clustering. Furthermore, the optimal α for BCS is computed as

$$\alpha = \frac{1}{2}\log\frac{\sum_{i,j=1}^{n} p_{i,j}\delta(\Delta_{i,j},1)}{c\sum_{i,j=1}^{n} q_{i,j}\delta(\Delta_{i,j},1)} \tag{4.24}$$

Figure 4.3 summarizes the BCS algorithm. Finally, we can show that the objective function of BCS decreases exponentially, i.e.,

Theorem 2 *Let $\Delta^1, \Delta^2, \ldots, \Delta^T$ be the kernel matrices computed from the clustering results by running the boosting algorithm (in Figure 4.3). Then, the objective function after T iterations, i.e., $\mathcal{L}_T^{BCS}$, is bounded as follows:*

$$\mathcal{L}_T^{BCS} \leq \left(\sum_{i,j=1}^{n} S_{i,j}^{+} + cS_{i,j}^{-}\right)\prod_{t=1}^{T}(1-\gamma_t^s), \tag{4.25}$$

where

$$\gamma_t^s = \frac{(\sqrt{B_t}-\sqrt{D_t})^2}{A_t+B_t+C_t+D_t}$$

A_t, B_t, C_t, and D_t are already defined in Theorem 1.

4.4 Experiments

We now present an empirical evaluation of our proposed boosting framework. In particular, we aim to address the following four questions in our study:

1. *As a general boosting framework, is the proposed method able to improve the performance for any given clustering algorithm?*

2. *How effective is the proposed boosting framework in improving the clustering performance by using the pairwise constraints?*

3. *How robust is the proposed boosting framework in improving the clustering performance by using the pairwise constraints?*

4. *How does the BCP algorithm compare to BCS algorithm in the proposed boosting framework?*

4.4.1 Experiment Setup

To validate the claim that the proposed boosting algorithm is capable of improving *any* clustering algorithm by exploiting the pairwise constraints, three typical clustering algorithms are used in our study. They are:

1. K-means algorithm [5]. It represents the family of clustering algorithms that try to find compact and well-separated clusters. We adopted the implementation from the Weka software[2].

2. Partitional SingleLink algorithm ("SLINK" for short) [80]. It represents the family of the hierarchical clustering algorithms. We adopted the implementation from the CLUTO software[3]

3. k-way spectral clustering ("SPEC" for short). It represents the family of spectral methods for data clustering. In particular, we follow the paper [115] for the implementation of spectral clustering.

[2] `http://www.cs.waikato.ac.nz/ml/weka/`
[3] `http://glaros.dtc.umn.edu/gkhome/views/cluto`

Name	#Attributes	#Clusters	#Examples
wdbc	30	2	569
scale	4	3	625
vowel	10	11	990
segmentation	19	7	2310
handwrittiendigit	256	10	4000
pendigit	16	4	4396

Table 4.2. Datasets used in the experiments.

Six datasets drawn from the UCI machine learning repository [50] are used in our study. Table 4.2 summarizes the information about these datasets[4]. As indicated in Table 4.2, these datasets vary significantly in their sizes, number of clusters, and number of attributes.

To evaluate the clustering performance, two measurements are used in our experiments. The first measurement is normalized mutual information (**NMI** for short) [15], which is defined as

$$NMI = \frac{2MI(X, X_0)}{H(X) + H(X_0)}$$

where X_0 and X denote the random variables of cluster memberships from the ground truth and the output of clustering algorithm, respectively. $MI(x, y)$ represents the mutual information between random variables x and y, and $H(x)$ represents the Shannon entropy of random variable x. The second measurement is Pairwise F-measure (**PWF1** for short), which is the harmonic mean of pairwise *precision* and *recall* that are defined as follows

$$\begin{aligned} precision &= \frac{\#\text{pairs correctly placed in the same cluster}}{\text{Total } \#\text{pairs placed in the same cluster}} \\ recall &= \frac{\#\text{pairs correctly placed in the same cluster}}{\text{Total } \#\text{pairs actually in the same cluster}} \\ PWF1 &= \frac{2 \times precision \times recall}{precision + recall} \end{aligned}$$

[4]Note that for the "pendigit" dataset, examples in only four classes of letter "3", "6", "8" and "9" are selected from a total of 10 classes because these four letters are in general difficult to distinguish.

The PWF1 measurement defined above is closely related to the metric defined in [155] that measures the percentage of data pairs correctly clustered together. The key problem with the metric defined in [155] is that it counts two types of data pairs, i.e., pairs of data points assigned to the same cluster and pairs of data points assigned to different clusters, with equal importance. This is problematic because most of the data pairs will consist of data points from different clusters when the number of clusters is large. A similar issue was raised in multi-class learning, and that is why F-measure is widely used for evaluating multi-class learning [161].

To verify the efficacy of the proposed boosting framework in exploiting the pairwise constraints for data clustering, three baseline approaches are used:

1. *Metric Pairwise Constraints K-means* (**MPCKmeans** for short) algorithm [?, 12], which is a probabilistic framework based on Hidden Markov Random Fields.
2. *Semi-supervised Kernel K-means* (**SSKK** for short) algorithm [96], which exploits the pairwise constraints by a kernel approach and finds clusters with nonlinear boundaries in the input data space.
3. *Spectral Learning* (**SpectralLearn** for short) algorithm [89], which applies spectral methods to learn a data representation from the pairwise constraints. The generated data representation can therefore be used by *any* clustering algorithm to identify the appropriate data clusters. The key difference between spectral learning and our algorithm is that our algorithm generates algorithm specific data representations by taking into account the performance of clustering algorithms.

Previous studies [15, 12, 96, 89] showed that the above three algorithms deliver the state-of-the-art performance in comparison to other semi-supervised clustering algorithms such as the constrained K-means.

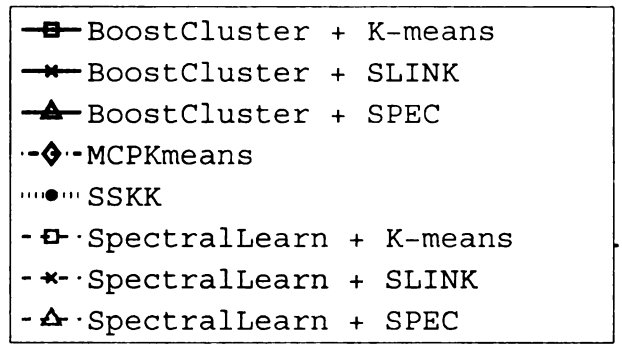


Figure 4.5. Legends for all algorithms in our comparative study. These legends will be used in all the figures in this paper.

In summary, we will compare the following semi-supervised clustering algorithms in the experiments: the three clustering algorithms (K-means, SLINK, and SPEC) being boosted by the proposed BoostCluster framework; the same three clustering algorithms with input from the Spectral Learning algorithm; the MPCKmeans algorithm; and the SSKK algorithm. For easy identification in figures, we listed the legends for the above algorithms to be compared, in Figure 4.5. These legends apply to all following performance comparison figures (we will omit showing legends in those figures due to space constraints).

Finally, in all the experiments, we vary the number of pairwise constraints from 0 to 800. Since a random sampling of data pairs tends to find many more cannot-link pairs than the must-link pairs, in this study, we enforce an equal number of constraints for both must-link pairs and cannot-link pairs. The numbers of eigenvectors (i.e., the parameter s in the boosting algorithm shown in Figure 4.3) are determined empirically as follows: 3 for the "scale" dataset, 10 for the "handwrittendigit" dataset and 5 for the remaining 4 datasets. All the experiments in this study are repeated five times, and the evaluation results averaged over the five trials are reported.

4.4.2 Generality of the Boosting Framework

Figures 4.6 - 4.9 show the clustering performance, evaluated by NMI and PWF1 respectively, of the BCP algorithm of BoostCluster framework using the three clustering algorithms (i.e., K-means, partitional SingleLink, and spectral clustering), the same three clustering algorithms with input as the new data representation from the Spectral Learning algorithm, the MPCKmeans algorithm, and the SSKK algorithm.

1. We observe that for most datasets, the BoostCluster framework is able to improve the clustering performance for all the three clustering algorithms regardless of which evaluation metric is used. This suggests that the proposed framework is effective in exploiting the pairwise constraints to improve clustering performance. MPCKmeans algorithm and SSKK algorithm are also effective in general, however, their clustering performance improvements are less significant, especially for larger datasets (such as "handwrittendigit" and "pendigit").

2. Although SpectralLearn algorithm can also be combined with *any* clustering algorithm, in our experiments, it does not always improve the clustering algorithm performance. For example, for "wdbc" and "handwrittendigit", increasing the number of pairwise constraints deteriorates clustering performance by combining SpectralLearn with any of the three clustering algorithms. Moreover, the effect of SpectralLearning depends on the clustering algorithm. For example, for the "pendigit" dataset, SpectralLearn improves the clustering performance of K-means and SLINK, but degrades SPEC in general. In comparison, the clustering performance improvement brought by the proposed BoostCluster is substantially more stable and consistent, across different datasets and different clustering algorithms. This can be attributed to the fact that BoostCluster is adaptive to both clustering algorithms and datasets: in each iteration, it takes the feedback from the result of applying the given clustering algorithm to the

particular dataset, and decides how to adjust the kernel matrix. However, SpectralLearn generates new data representations independent from the clustering algorithm that is used.

3. The performance of the three clustering algorithms (K-means, SLINK, and SPEC) varies significantly across the six different datasets. For instance, for the "vowel" dataset, "BoostCluster+K-means" algorithm performs considerably worse than "BoostCluster+SPEC" algorithm. However, the performance of "BoostCluster+K-means" algorithm for the "pendigit" dataset, is significantly better than that of "BoostCluster+SPEC" algorithm. This result also indicates that every clustering algorithm has its own strength, and therefore it is important to develop a general framework that is able to boost the performance of any clustering algorithm by the given pairwise constraints.

4. The results based on the two different evaluation metrics, namely NMI and PWF1, are inconsistent on some occasions. For instance, for the "handwrittingdigit" dataset, according to NMI, the clustering performance of "BoostCluster+K-means" and "BoostCluster+SLINK" appears to be similar. However, according to PWF1, "BoostCluster+SLINK" performs noticeably better than "BoostCluster+K-means". The implication of this finding is the importance of evaluating clustering performance by more than one evaluation metric, since conclusions based on the results of a single evaluation metric could be biased.

4.4.3 Robustness of Exploiting Pairwise Constraints

Although the curves in Figure 4.6 - Figure 4.9 all display different degrees of "bumpiness", generally speaking, BoostCluster framework, SSKK and MPCKmeans deliver a more robust performance than SpectralLearn algorithm. On the other hand, although for most datasets, the performance curves of SSKK appear to be the most

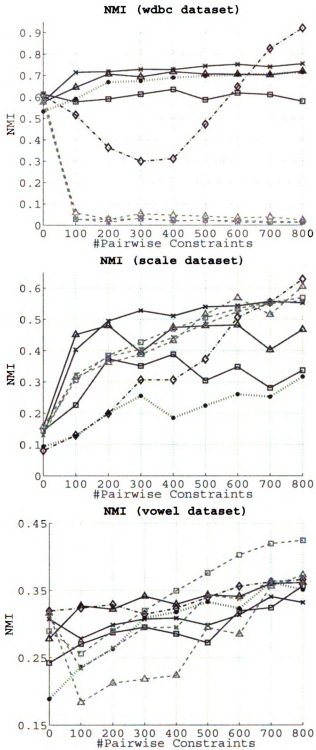


Figure 4.6. Clustering performance evaluated by NMI (Part A). Each graph shows the performance of the three clustering algorithms (K-means, parititional SLINK, spectral clustering) boosted by the proposed BCP algorithm, and the performance of the MPCKmeans, SSKK and SpectralLearn algorithms.

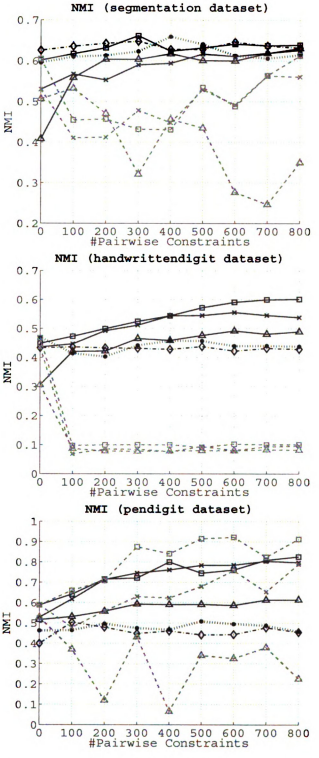


Figure 4.7. Clustering performance evaluated by NMI (Part B). Each graph shows the performance of the three clustering algorithms (K-means, parititional SLINK, spectral clustering) boosted by the proposed BCP algorithm, and the performance of the MPCKmeans, SSKK and SpectralLearn algorithms.

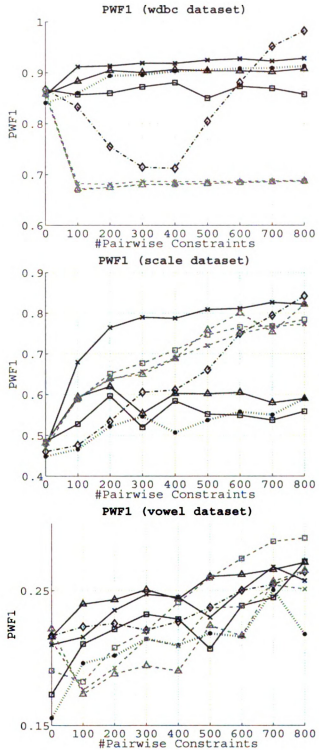


Figure 4.8. Clustering performance evaluated by PWF1 (Part A). Each graph shows the performance of the three clustering algorithms (K-means, partitional SLINK, spectral clustering) boosted by the proposed BCP algorithm, and the performance of the MPCKmeans, SSKK and SpectralLearn algorithms.

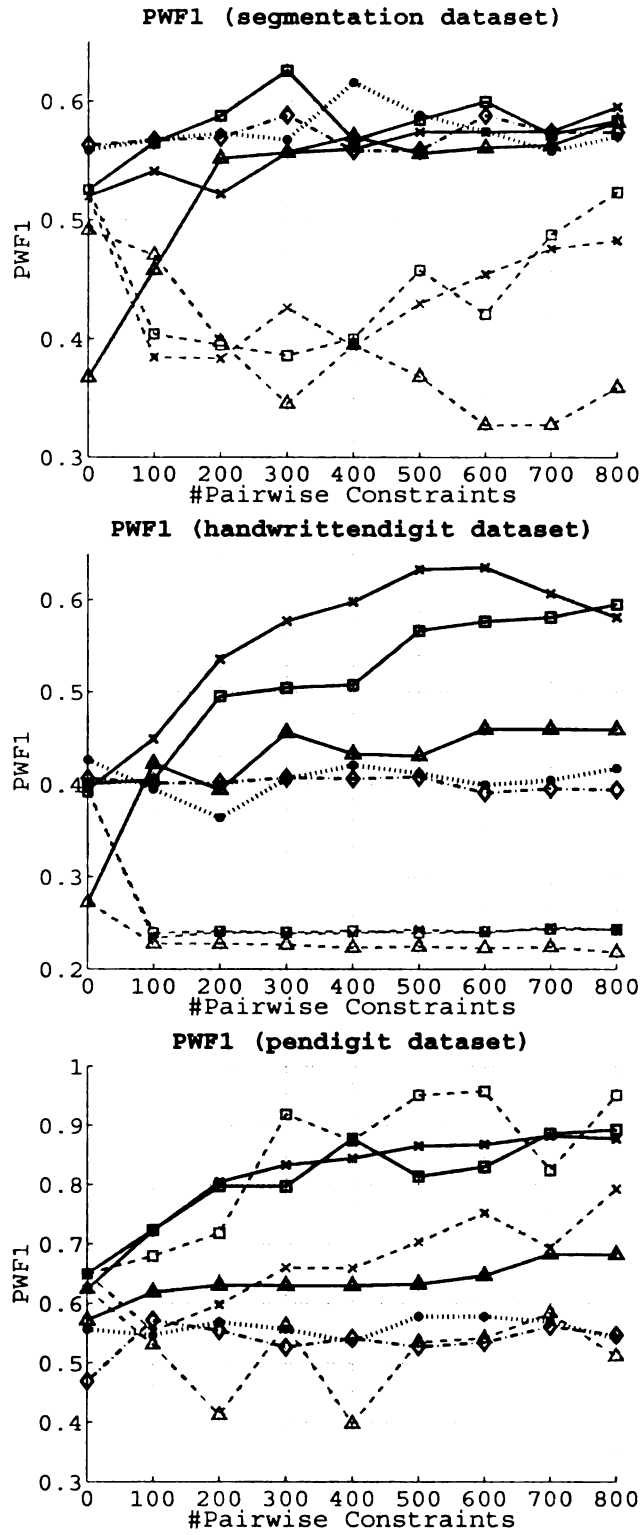


Figure 4.9. Clustering performance evaluated by PWF1 (Part B). Each graph shows the performance of the three clustering algorithms (K-means, partitional SLINK, spectral clustering) boosted by the proposed BCP algorithm, and the performance of the MPCKmeans, SSKK and SpectralLearn algorithms.

smooth among all the competitors, the resultant improvement is almost always the least noticeable among all the semi-supervised clustering algorithms.

To further evaluate the robustness of all the algorithms, we conduct experiments with noisy pairwise constraints. We randomly select 20% of the pairwise constraints and flip their labels (i.e., a must-link pair becomes a cannot-link pair and vice versa). This setting reflects the scenario when the side information includes incorrect pairwise constraints. It could happen when for instance, the pairwise constraints are derived from the implicit user feedback (e.g., user ratings or click-through data). Thus, it is important to develop semi-supervised clustering algorithms that are resilient to the noisy side information.

Figure 4.10 and Figure 4.11 show the performance of all the algorithms, on three selected datasets (i.e., "scale", "vowel", and "pendigit") when 20% of the pairwise constraints are noisy. First, by comparing Figure 4.10 - Figure 4.11 with Figures 4.6 - Figure 4.9, it is not surprising to observe a degradation in clustering performance when 20% of the pairwise constraints are noisy. Second, we observe a general trend that a larger number of noisy constraints tend to result in an inferior clustering performance by MPCKmeans. This is in contrast to the results of MPCKmeans shown in Figure 4.6 - Figure 4.9 where increasing the number of pairwise constraints usually improves the performance of clustering. This result implies that the MPCKmeans algorithm is unable to effectively exploit the pairwise constraints for data clustering when they are noisy. Similarly, while "SpectralLearn+K-means" and "SpectralLearn+SLINK" are able to noticeably improve the clustering performance with increasing number of noise-free pairwise constraints, with 20% noise in the constraints, their performance degrades with the increasing number of constraints. In comparison, as shown in Figure 4.10 and Figure 4.11, BoostCluster framework is generally able to improve the performance of all the three clustering algorithms with increasing number of noisy pairwise constraints, and SSKK algorithm is also able to improve clustering perfor-

DATASETS	K-means (BCP)		K-means (BCS)	
	NMI	PWF1	NMI	PWF1
wdbc	0.5862	0.8502	0.6360	0.8221
scale	0.3040	0.5520	0.3450	0.5645
vowel	0.2723	0.2069	0.3317	0.2565
segmentation	0.6315	0.5843	0.6013	0.5544
handwrittendigit	0.5720	0.5664	0.5435	0.5359
pendigit	0.7448	0.8140	0.7261	0.8166
DATASETS	SLINK (BCP)		SLINK (BCS)	
	NMI	PWF1	NMI	PWF1
wdbc	0.7438	0.9248	0.7971	0.9440
scale	0.5399	0.8091	0.5015	0.7839
vowel	0.2982	0.2302	0.3044	0.2599
segmentation	0.6164	0.5740	0.5765	0.5421
handwrittendigit	0.5451	0.6328	0.5300	0.5579
pendigit	0.7833	0.8654	0.7713	0.8573
DATASETS	SPEC (BCP)		SPEC(BCS)	
	NMI	PWF1	NMI	PWF1
wdbc	0.7073	0.9038	0.6749	0.8964
scale	0.4792	0.6021	0.3818	0.6099
vowel	0.3431	0.2606	0.3417	0.2649
segmentation	0.6004	0.5559	0.5754	0.5768
handwrittendigit	0.4764	0.4306	0.4579	0.4417
pendigit	0.5912	0.6325	0.8056	0.8745

Table 4.3. The performance comparison between the BCP algorithm and the BCS algorithm, when the number of pairwise constraints is 500.

mance despite the noise in pairwise constraints. This indicates that the proposed BoostCluster framework and the SSKK algorithm are more resilient to the noise in the side information.

4.4.4 BCP vs. BCS

We compare the clustering performance of the BCP algorithm and the BCS algorithm in Table 4.3 that is evaluated in both NMI and PWF1. Due to the space limitation, we only list the evaluation results for 500 pairwise constraints. We set $c = 1$ in the objective function (4.7) of the BCS algorithm.

As indicated in Table 4.3, in general the BCP and BCS algorithms deliver similar performance, which suggests that both algorithms are effective in boosting the performance of the underlying clustering algorithms. BCP is more effective than BCS for certain datasets and clustering algorithms, and vice versa. However, it is important

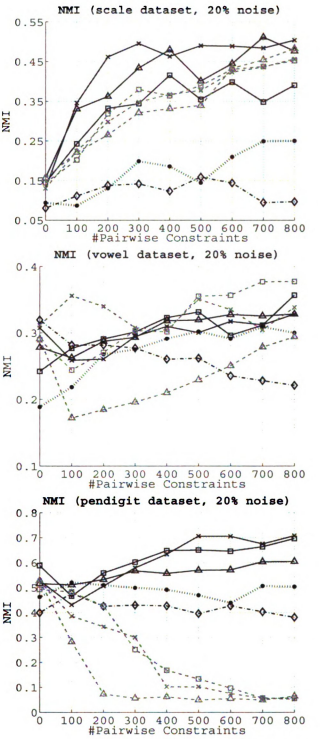


Figure 4.10. Clustering performance (NMI) with 20% noise in the pairwise constraints. Each graph shows the performance of the three clustering algorithms (K-means, paritional SLINK, spectral clustering) boosted by the proposed BCP algorithm of BoostCluster framework, and the performance of the MPCKmeans, SSKK and SpectralLearn algorithms.

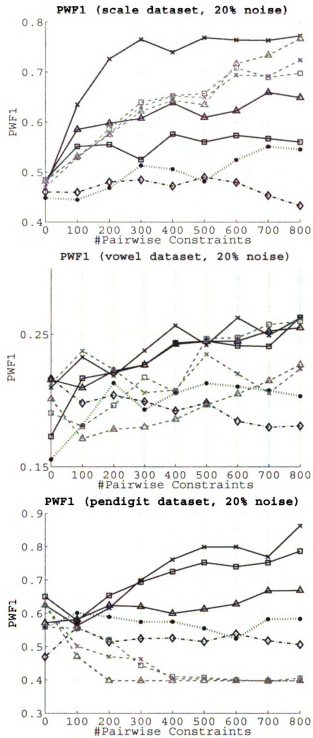


Figure 4.11. Clustering performance (PWF1) with 20% noise in the pairwise constraints. Each graph shows the performance of the three clustering algorithms (K-means, paritional SLINK, spectral clustering) boosted by the proposed BCP algorithm of BoostCluster framework, and the performance of the MPCKmeans, SSKK and SpectralLearn algorithms.

to note that compared to BCP, BCS has an additional parameter c that can be used to tune the boosting algorithm. Our empirical experience indicates that changing the parameter c can lead to significant difference in the clustering performance. It would be useful to investigate the pros and cons of the BCP and BCS algorithms, and how to choose the parameter c in the BCS objective function wisely.

4.5 Summary

In this chapter, we have studied the problem of improving data clustering by using side information in the form of pairwise constraints. A general boosting framework has been proposed to improve the accuracy of *any* given clustering algorithm with a given set of pairwise constraints. Such performance improvement is achieved by iteratively finding new data representations that are consistent with both the clustering results from previous iterations and the pairwise constraints. Empirical study shows that our proposed boosting framework is able to improve the clustering performance of several popular clustering algorithms by using the pairwise constraints.

CHAPTER 5

Semi-supervised Learning for Query Translation Disambiguation in Dictionary-based Cross Language Information Retrieval

5.1 Introduction

To bridge the gap between different languages, machine translation has been used extensively in many research areas of multilingual information processing, such as cross-language information retrieval (CLIR). In CLIR, we can either translate queries into the language of documents or translate documents into the language of queries. Usually, it is simpler and more efficient to translate queries because of their shorter length. Most query translation algorithms require external linguistic resources, among which parallel corpora and bilingual dictionaries are the most commonly used. Methods based on parallel corpora usually learn the association between words of the source language and words of the target language, and apply the learned association to estimate the translation of queries. Examples in this category include statistical translation models [157, 53, 116, 94], and relevance language models [93, 101, 100]. The main drawback of these methods is that they depend critically on the availability

of parallel bilingual corpora, which are often difficult to acquire, especially for minor languages.

While the problem of machine learning for language translation remains tough (either theoretically or practically), undoubtedly, bilingual dictionaries can be viewed as side information for various multi-lingual tasks. Typically, a bilingual dictionary provides a repository of translation pairs, usually organized in a manner that each word (or phrase) in one language is supplied with a list of possible translation words (or phrases) in another language. These translation pairs, to some extent, helps to bridge (or narrow) the gap between languages. With the increasing availability of machine readable bilingual dictionaries, dictionary-based approaches becomes more preferable for CLIR applications, especially when other linguistic resources are scarce.

Compared to the approaches based on parallel corpora, a major disadvantage of the bilingual dictionary based approaches is that they lack the ability in disambiguating the translation of query terms among multiple candidates. Very often, a number of translations (which we call *translation candidates* in this chapter) are found in a bilingual dictionary for a single query word, but most of them are irrelevant to the semantic meaning of the query. Hence, it is crucial for a dictionary-based approach to reduce the ambiguity in translating query words as much as possible. However in CLIR, given the short length of a query, it is usually impossible to completely resolve the translation ambiguity due to the multiple interpretation of the query. Thus, it is also important for any dictionary-based CLIR approach to maintain the uncertainty in translating queries when the ambiguity is hard to resolve.

In the past, several approaches [81, 57, 58, 1, 95] have been proposed to resolve the query translation ambiguity in dictionary-based CLIR. The simplest one is to use all the translation candidates of each query word provided by the dictionary with equal weights [46, 95]. This amounts to no sense disambiguation when translating query words. Other approaches try to resolve the translation ambiguity by measuring the

coherence of a translation candidate to the entire query. Typically, the coherence score of a translation candidate is computed using word co-occurrence statistics. Given a query, a translation candidate of a query word is assigned with a high coherence score when it co-occurs frequently with the translations of other query words. The translation candidates with the highest coherence scores are selected to form the final translation for the original query. In [46, 76, 2, 93, 95, 57], only one translation candidate is selected for each query word; in [81, 109], a translation candidate is selected when its coherence score exceeds a predefined threshold, which allows multiple translations to be selected for each query word. We will refer to both approaches mentioned above as *selection-based approaches*, because they all have to make a *binary* decision for each translation candidate regarding if it will be included in the translated query or not. Given the usually short length of queries and the large variance existed in mapping information across different languages, such binary decisions are usually difficult, if not impossible, to make. We call this problem the "***translation uncertainty problem***". Another problem with the selection-based approaches is that the translation of one query word is usually determined independently from the translations of others. This assumption is reflected in the calculation of coherence scores. Usually, the coherence score of a translation candidate to a given query is computed as the sum of its similarities to *every* translation candidate for other query words. As a result, coherence scores are estimated independently from the choice of translations for query words, which leads the selection of translation candidates for different query words to be independent. We call this problem "***translation independence assumption***". Although this problem has been addressed in previous work (e.g., [57]), usually greedy approaches are applied and therefore only suboptimal solutions can be obtained.

In this chapter, we propose a novel statistical framework for dictionary-based CLIR. This framework will allow us to estimate the translation probabilities of query

words by maximizing the overall coherence of the translated query, which we call "***maximum coherence principle***". Particularly, the proposed framework explicitly addresses the two problems mentioned above: to resolve the translation uncertainty problem, the proposed framework maintains the uncertainty in translating queries through the estimation of translation probabilities of query words; to remove the translation independence assumption, the proposed framework allows the translation probabilities of *all* query words to be estimated simultaneously. Furthermore, the proposed framework is formulated as a quadratic programming problem [62], whose global optimal solution can be found efficiently using standard optimization packages such as Matlab. This is in contrast to several existing approaches such as the propagation approach in [114], where the solution is determined by an iterative procedure, which is not only time consuming but also sensitive to the initialization of parameters or the stop criterion employed in the iterative procedure.

In addition to the general framework, we also present in this chapter two realizations of the proposed framework that employ different coherence measurements: the "**Maximum Coherence Model**" that adopts the *raw* word-to-word similarity for coherence measurement, and the "**Spectral Query Translation Model**" that measures the coherence score of each translation candidate based on the *normalized* word similarity. As will be explained later, the Spectral Query Translation Model based on the normalized similarity measurement, can be further explained as a graph partitioning approach for query translation disambiguation, which is employed in spectral clustering. Our empirical studies with TREC datasets have shown that both models outperform the selection-based approaches with relative improvements ranging from 10% to 50%.

5.2 Related Work

We first review the previous work in the selection-based approaches for query translation disambiguation, followed by the discussion of spectral clustering that is strongly related to the proposed Spectral Query Translation Model.

5.2.1 Selection-based Approaches for Query Translation Disambiguation

One of the major factors that can potentially degrade the effectiveness of dictionary-based cross-language information retrieval is the ambiguity in translating query words [8, 57]. In the efforts to resolve this translation ambiguity, a number of recent studies [46, 76, 81, 2, 93, 109, 57, 95] have suggested the strategy of translation selection by exploiting word co-occurrence patterns. Usually a similarity measurement between two translation candidates is defined in the form of word co-occurrence statistics. With the word similarities, we can then measure the coherence of a translation candidate with regard to the theme of the entire query. Only those translation candidates with high coherence scores will be selected for the query translation.

Ideally, for each query word we should select the translation candidate(s) that is consistent with the selected translation candidates for other query words. Apparently, this becomes a "chicken-egg" problem since the selection of translation candidates for one query word is determined by the translation candidates selected for other query words. Thus, due to the computational concern, most selection-based approaches [1, 57, 58] adopted approximate approaches that usually only produce suboptimal solutions. In particular, for each query word, those approaches choose the translation candidates that are consistent with *all* the translation candidates provided by the dictionary for all the query words, including both the selected and the unselected translation candidates. Formally, a translation selection strategy can be formulated as follows:

APPROXIMATE TRANSLATION SELECTION ALGORITHM

1. Given a query $\mathbf{q}^s = \{q_1^s, q_2^s, \cdots, q_{m^s}^s\}$ in the source language , for each query word q_i^s, look up the dictionary for the translation candidate set $S_i = \{w_{i,j}^t\}$

2. For each set S_i

   (a) For each translation candidate $w_{i,j}^t$ in S_i, define the similarity measurement between the word $w_{i,j}^t$ and the set $S_{i'}(i' \neq i)$ as the sum of the similarities between $w_{i,j}^t$ and each word in the set $S_{i'}$, i.e.,

   $$sim(w_{i,j}^t, S_{i'}) = \sum_{\forall w_{i',l}^t \in S_{i'}} sim(w_{i,j}^t, w_{i',l}^t) \tag{5.1}$$

   where $sim(w_{i,j}^t, w_{i',l}^t)$ computes the word-to-word similarity.

   (b) Compute the coherence score for $w_{i,j}^t$ as

   $$f(w_{i,j}^t) = \sum_{\forall i' \neq i} sim(w_{i,j}^t, S_{i'}) \tag{5.2}$$

   (c) Select the word q_i^t in S_i with the highest coherence score

   $$q_i^t = \arg\max_{w_{i,j}^t} f(w_{i,j}^t) \tag{5.3}$$

The definition of similarity between two words in the above algorithm can take various forms of co-occurrence statistics, such as Dice similarity (as in [1]), mutual information (as in [81, 109]) or its variants (as in [57, 58]). In addition to selecting the most likely translation for each query word, other selection-based approaches have been tried, such as selecting the best N translations [46] or selecting translations whose coherence scores exceed a predefined threshold [81, 109].

Apparently the above approximate algorithm is not ideal. In particular, the coherence score for a translation is computed with regard to both selected and unselected translations. As a result of such an approximation, translation of different query

words are determined independently, which leads to the translation independence problem as discussed in the introduction section. In the proposed statistical framework, by formulating the problem of translation selection in a quadratic programming form, we are able to efficiently estimate the translations of *all* query words *simultaneously*. Furthermore, in contrast to the selection-based approaches that make binary decision for each translation candidate, the new framework employs soft probabilities for representing both selected and unselected translation candidates. This is particularly useful when binary decisions are hard to make, for instance, all the translation candidates of a query word have very similar coherence scores.

5.2.2 Spectral Clustering

Spectral clustering approaches view the problem of data clustering as a problem of graph partitioning. Each data point corresponds to a vertex in a graph. Any two data points are connected by an edge whose weight is the similarity between the two data points. To form data clusters, the graph is partitioned into multiple disjoint sets such that only the edges with small weights are removed. Based on different criteria imposed on the partitioning, there are three major variants for spectral clustering: Ratio Cut [38], Normalized Cut [134] and Min-Max Cut [49]. In the following, we briefly recapitulate the 2-way Normalized Cut algorithm since it is the most widely used spectral clustering algorithm and has the closest relation to our proposed work.

Let $G(V, E; \mathbf{W})$ denote an undirected graph, where V is the vertex set, E is the edge set, and $\mathbf{W} = (w_{i,j})_{n \times n}$ is a matrix with $w_{i,j} \geq 0$ denoting the edge weight between the i-th and the j-th vertex. Define $\mathbf{D} = diag(d_1, d_2, \cdots, d_n)$, where $d_i = \sum_{j \in V} w_{i,j}$. To partition the vertex set into two disjoint sets A and B, a 2-way Normalized Cut algorithm minimizes the following objective function:

$$J = \frac{S(A,B)}{d_A} + \frac{S(A,B)}{d_B} \tag{5.4}$$

where we define $S(A,B) = \sum_{i \in A} \sum_{j \in B} w_{i,j}$ as the *cut* value, $d_A = \sum_{i \in A} d_i$ and

$d_B = \sum_{i \in B} d_i$ as normalizing factors that balance the size of the two clusters. The above objective function can be rewritten as

$$J = \sum_{i \in A} \sum_{j \in B} w_{i,j} \frac{d_A + d_B}{d_A d_B} = \sum_{i \in A} \sum_{j \in B} w_{i,j} \frac{(d_A + d_B)^2}{d_A d_B (d_A + d_B)} \tag{5.5}$$

If we introduce a cluster indicator vector $\mathbf{q}$, with each element q_i defined as

$$q_i = \begin{cases} \sqrt{d_B / [d_A (d_A + d_B)]} & \text{if } i \in A \\ \sqrt{d_A / [d_B (d_A + d_B)]} & \text{if } i \in B \end{cases} \tag{5.6}$$

the objective function becomes

$$\begin{aligned} J &= \sum_{i,j \in V} (q_i - q_j)^2 w_{i,j} \\ &= \sum_{i,j \in V} (q_i^2 + q_j^2 - 2 q_i q_j) w_{i,j} \\ &= \sum_{i \in V} 2 q_i^2 (\sum_{j \in V} w_{i,j}) - \sum_{i,j \in V} 2 q_i q_j w_{i,j} \\ &= 2 \sum_{i \in V} q_i^2 d_i - 2 \sum_{i,j \in V} q_i q_j w_{i,j} \\ &= 2 \mathbf{q}^\top (\mathbf{D} - \mathbf{W}) \mathbf{q} \end{aligned} \tag{5.7}$$

Note that the minimizer to the above normalized cut value J is a binary vector $\mathbf{q}$, with each element q_i indicating the cluster membership of a vertex. Given its combinatorial nature, it is difficult to solve the optimization problem efficiently (NP hard). However, if we relax the cluster memberships to real values under the constraints

$$\mathbf{q}^\top \mathbf{D} \mathbf{q} = 1 \tag{5.8}$$

$$\mathbf{q}^\top \mathbf{D} \mathbf{e} = 0 \quad (\text{where } \mathbf{e} = [1, \cdots, 1]^\top) \tag{5.9}$$

the Normalized Cut algorithm can be formulated as follows:

$$\begin{aligned} &\min_{\mathbf{q}} \ \mathbf{q}^\top (\mathbf{D} - \mathbf{W}) \mathbf{q} \\ &s.t. \quad \mathbf{q}^\top \mathbf{D} \mathbf{q} = 1, \quad \mathbf{q}^\top \mathbf{D} \mathbf{e} = 0 \end{aligned} \tag{5.10}$$

Furthermore, if we define $\tilde{\mathbf{q}} = \mathbf{D}^{\frac{1}{2}}\mathbf{q}$, we reach the following equivalent optimization problem

$$\begin{aligned} \min_{\tilde{\mathbf{q}}} \quad & \tilde{\mathbf{q}}^\top(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}})\tilde{\mathbf{q}} \\ s.t. \quad & \tilde{\mathbf{q}}^\top\tilde{\mathbf{q}} = 1, \quad \tilde{\mathbf{q}}^\top\mathbf{D}^{\frac{1}{2}}\mathbf{e} = 0 \end{aligned} \tag{5.11}$$

Note the above problem is in the form of Rayleigh quotient, and its solution can be found by solving the following eigenvalue system [65]

$$(\mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}})\tilde{\mathbf{q}} = \lambda\tilde{\mathbf{q}} \tag{5.12}$$

5.3 The Statistical Framework For Dictionary-based CLIR

The essential idea of the framework is to learn a set of translation probabilities for query words from the word co-occurrence statistics that maximizes the overall coherence of the translated query. In the following subsections, we will describe the components of the general statistical framework for dictionary-based cross-language information retrieval, including uncertainty modeling, the retrieval model, translation probabilities learning and the solution to the related optimization problem, followed by a summary that sketches the steps of applying the proposed framework to CLIR.

5.3.1 Notation

The term "source language" and a superscript s are used when referring to the language of queries. Similarly, the term "target language" and a superscript t are for the language of documents. Let a query of the source language be denoted by $\mathbf{q}^s = \{w_1^s, w_2^s, \cdots, w_{m^s}^s\}$, where m^s is the number of distinct words in $\mathbf{q}^s$. Let $\mathbf{r}_k$ denote the set of translation candidates provided by the dictionary for a word w_k^s in the source language. The union of translation candidate sets for all the words in $\mathbf{q}^s$ is then denoted by $\mathbf{R} = \bigcup_{k=1}^{m^s} \mathbf{r}_k$. The total number of distinct translation candidates

for query $\mathbf{q}^s$, i.e., the size of $\mathbf{R}$, is denoted by m^t . A matrix $\mathbf{T} = [t_{k,j}]_{m^s \times m^t}$ represents the part of the bilingual dictionary related to query $\mathbf{q}^s$. Each element $t_{k,j}$ in $\mathbf{T}$ is 1 if the j-th word in the target language appears in the dictionary as a translation for the k-th word in the source language, and 0 otherwise.

5.3.2 Modeling the Uncertainty in Query Translation

To address the problem of translation uncertainty, we build the statistical framework by introducing translation probabilities. For a given query word, instead of making binary decision for its translation candidates, we estimate the *probability* of translating the query word into each translation candidate. More importantly, the translation probabilities are estimated under the context of the entire query, namely a translation candidate will be assigned large probability mass if it is coherent with the semantic meaning of the entire query and vice versa.

Let $p_{k,j}$ denote the probability of translating a word w_k^s of the source language into a word w_j^t of the target language, given the context of query $\mathbf{q}^s$. It is defined as

$$p_{k,j} = \Pr(w_j^t | w_k^s, \mathbf{q}^s) \tag{5.13}$$

In order to be consistent with the dictionary $\mathbf{T}$, we assume that translation probability $p_{k,j} = 0$ if the word w_j^t does not appear in the dictionary as the translation of word w_k^s. In other words, $p_{k,j}$ could be nonzero only if the word w_j^t is one of the translation candidates for the word w_k^s. This assumption can be formally expressed by the following constraints:

$$\forall k = 1, ..., m_s,\ j = 1, ..., m_t : \quad 0 \leq p_{k,j} \leq t_{k,j}$$

In addition, we have a constraint

$$\forall k = 1, ..., m_s : \sum_{\forall w_j^t \in \mathbf{r}_k} p_{k,j} = 1$$

to ensure that each query word has only one ideal translation given the context of the query $\mathbf{q}^s$.

To simplify our notation, we further introduce the matrix $\mathbf{P} = \left[p_{k,j}\right]_{m^s \times m^t}$ to denote all the translation probabilities for query $\mathbf{q}^s$. Then, the above two sets of constraints can be rewritten as

$$\mathbf{P} \cdot \mathbf{e}_{m^t \times 1} = \mathbf{e}_{m^s \times 1} \tag{5.14}$$

$$\mathbf{0} \leq \mathbf{P} \leq \mathbf{T} \tag{5.15}$$

where $\mathbf{e} = [1, 1, \ldots, 1]^\top$.

5.3.3 The Retrieval Model

The introduction of translation probabilities $p_{k,j}$ can be well accommodated by a statistical retrieval model for CLIR. In particular, we estimate $\Pr(\mathbf{d}^t|\mathbf{q}^s)$, i.e., the probability for a document $\mathbf{d}^t$ in the target language to be relevant to a query $\mathbf{q}^s$ in the source language. By the Bayes' law, this probability can be approximated as

$$\Pr(\mathbf{d}^t|\mathbf{q}^s) = \frac{\Pr(\mathbf{q}^s|\mathbf{d}^t) \cdot \Pr(\mathbf{d}^t)}{\Pr(\mathbf{q}^s)} \sim \Pr(\mathbf{q}^s|\mathbf{d}^t) \tag{5.16}$$

The last step assumes that document prior $\Pr(\mathbf{d}^t)$ follows a uniform distribution. Hence, in the following, we will compute $\log \Pr(\mathbf{q}^s|\mathbf{d}^t)$, instead of $\log \Pr(\mathbf{d}^t|\mathbf{q}^s)$.

To model the translation uncertainty, we rewrite the expression for $\log \Pr(\mathbf{q}^s|\mathbf{d}^t)$ as follows:

$$\begin{aligned}
\log \Pr(\mathbf{q}^s|\mathbf{d}^t) &= \log \int d\mathbf{q}^t \Pr(\mathbf{q}^s|\mathbf{q}^t) \Pr(\mathbf{q}^t|\mathbf{d}^t) \\
&\approx \frac{1}{\Pr(\mathbf{q}^s)} \int d\mathbf{q}^t \Pr(\mathbf{q}^s|\mathbf{q}^t) \left(\log \Pr(\mathbf{q}^t|\mathbf{d}^t) + \log \Pr(\mathbf{q}^s)\right) \\
&\sim \int d\mathbf{q}^t \Pr(\mathbf{q}^t|\mathbf{q}^s) \log \Pr(\mathbf{q}^t|\mathbf{d}^t) \\
&= \sum_{w^t} \langle \log \Pr(w^t|\mathbf{q}^s) \rangle_{\Pr(\mathbf{q}^t|\mathbf{q}^s)} \\
&= \sum_{w^t} \Pr(w^t|\mathbf{q}^s) \log \Pr(w^t|\mathbf{d}^t)
\end{aligned} \tag{5.17}$$

In the second step of the above derivation we employs the Jensen's inequality [125], which can be viewed as the first step toward the variational approximation [79, 88]. In the third step, we ignore the term $\log \Pr(\mathbf{q}^s)$ that is independent from the document $\mathbf{d}^t$, and we also switch the roles between $\mathbf{q}^t$ and $\mathbf{q}^s$ using the Bayes' law (similar to (5.16)) by assuming that the prior of the translated query $\mathbf{q}^t$ follows a uniform distribution, i.e. $\Pr(\mathbf{q}^t)$ is a constant. In the fourth step, $\langle \cdot \rangle$ represents the mathematical expectation of a random variable. To estimate $\Pr(w^t|\mathbf{q}^s)$, i.e., the probability of observing the word w^t in the translation of the query $\mathbf{q}^s$, we decompose $\Pr(w^t|\mathbf{q}^s)$ into a summation over words in the source language:

$$\Pr(w^t|\mathbf{q}^s) = \sum_{w^s \in \mathbf{q}^s} \Pr(w^t|w^s;\mathbf{q}^s) \Pr(w^s|\mathbf{q}^s) \tag{5.18}$$

Finally, from (5.16) - (5.18) we have

$$\log \Pr(\mathbf{d}^t|\mathbf{q}^s) \sim \sum_{w^t} \sum_{w^s} \Pr(w^t|w^s;\mathbf{q}^s) \Pr(w^s|\mathbf{q}^s) \log \Pr(w^t|\mathbf{d}^t) \tag{5.19}$$

Here $\Pr(w^t|\mathbf{d}^t)$ is a monolingual language model for a document $\mathbf{d}^t$ in the target language; $\Pr(w^t|w^s;\mathbf{q}^s)$ is the probability for translating a query word w^s into w^t given the context of query $\mathbf{q}^s$; and $\Pr(w^s|\mathbf{q}^s)$ is a monolingual language model for query $\mathbf{q}^s$ in the source language, which can also be seen as the weight assigned to the query word w^s. For the sake of simplicity, an uniform distribution is assumed for probability $\Pr(w^s|\mathbf{q}^s)$. As indicated in (5.19), the key component to the above retrieval model is how to estimate the translation probabilities $\Pr(w^t|w^s;\mathbf{q}^s)$.

5.3.4 Learning the Translation Probabilities

In this subsection, we will describe the essential part of the statistical framework, i.e., automatically learning translation probabilities from the word co-occurrence statistics. We will begin with the definition of an overall coherence measurement for translating a query, followed by formulating the learning process in an optimization form.

Using the translation probabilities introduced in the previous subsection, we can now define a measurement for the overall coherence when translating a query $\mathbf{q}^s$, i.e.,

$$Co(\mathbf{q}^s;\mathbf{T}) = \sum_{\substack{\forall w_k^s \in \mathbf{q}^s \\ \forall w_{k'}^s \in \mathbf{q}^s}} \sum_{\substack{\forall w_j^t \in \mathbf{r}_k \\ \forall w_{j'}^t \in \mathbf{r}_{k'}}} p_{k,j} \cdot o_{j,j'}^t \cdot p_{k',j'} \tag{5.20}$$

where $o_{j,j'}$ is a pair-wise similarity that measures the correlation between two words w_j^t and $w_{j'}^t$ in the target language.

The above measurement is motivated by the intuition that appropriate translations of query words tend to be coherent with each other. In other words, if w_j^t and $w_{j'}^t$ are appropriate translations for words w_k^s and $w_{k'}^s$ respectively, we expect that 1) both translation probabilities $p_{k,j}$ and $p_{k',j'}$ are assigned large values, and in the same time, 2) w_j^t and $w_{j'}^t$ are related by a large coherence measurement $o_{k,k'}^t$. Hence, what is implied by this intuition is that the assignment of probability $p_{k,j}$ and $p_{k',j'}$ should be synchronized with the coherence measurement $o_{j,j'}^t$. This synchronization is expressed in (5.20) through the multiplication of the three terms $p_{k,j}$, $p_{k',j'}$, and $o_{j,j'}^t$. In particular, by maximizing the overall coherence in (5.20), we enforce the consistency between the probability assignment and the coherence measurement by assigning large values to $p_{k,j}$ and $p_{k',j'}$ only when the corresponding coherence measurement $o_{j,j'}^t$ is large.

The similarity information (i.e., $o_{j,j'}^t$) can be derived from monolingual word co-occurrence statistics using the metric such as information gain. The expression for the overall coherence can be simplified using the matrix notation introduced in (5.14) and (5.15):

$$Co(\mathbf{q}^s;\mathbf{T}) = \mathbf{e}^\top \mathbf{P}\mathbf{O}\mathbf{P}^\top \mathbf{e} \tag{5.21}$$

where $\mathbf{O} = [o_{j,j'}^t]_{m^t \times m^t}$ is the similarity matrix that includes the similarity measurement of any two words in the target language.

It is important to note that the coherence measurement defined above is significantly different from the coherence measurement defined in [58]. The key difference between them lies in the fact that the measurement defined in [58] is based on the concept of translation selection, namely only the best translation candidate is chosen for every query word. As a consequence, the resulting formalization in [58] is indeed a combinatorial optimization problem, and therefore is difficult to solve efficiently. By relaxing the binary choice of translation for query words to translation probabilities, on one hand we resolve the difficulty in optimization, and on the other hand we are able to explore the translation uncertainty, which could be important when the translation ambiguities are difficult to resolve.

Now our goal is to determine the translation probabilities such that the overall coherence is maximized. Putting together both the objective function in (5.21) and the constraints in (5.14) and (5.15), the learning process of the query-dependent translation probabilities can be formalized as the following optimization form

$$\begin{aligned} \max_{\mathbf{P}\in\mathbb{R}^{m_s\times m_t}} \quad & \mathbf{e}^\top\mathbf{P}\mathbf{O}\mathbf{P}^\top\mathbf{e} - C_p\mathbf{e}^\top\mathbf{P}\mathbf{P}^\top\mathbf{e} \\ s.t. \quad & \mathbf{P}\cdot\mathbf{e}_{m^t\times 1} = \mathbf{e}_{m^s\times 1} \\ & \mathbf{0}\le\mathbf{P}\le\mathbf{T} \end{aligned} \tag{5.22}$$

Notice that in the above objective function, in addition to the first term that corresponds to the overall coherence measurement for query translation, another term $-C_p\mathbf{e}^\top\mathbf{P}\mathbf{P}^\top\mathbf{e}$ is introduced. This additional term is called a *regularizer* in machine learning, and plays the similar role as the prior in Bayesian learning [113]. Note that $\mathbf{e}^\top\mathbf{P}\mathbf{P}^\top\mathbf{e} = \sum_{k,k'}\sum_j p_{k,j}p_{k',j}$ stands for the sum of all elements in $\mathbf{P}\mathbf{P}^\top$, and its maximizer is to assign uniform distributions to all translation probabilities $\mathbf{P}$. Thus, by including the regularizer in the objective function, we essentially introduce an uninformative prior for $\mathbf{P}$, i.e., without the context of a given query, we assume that all translation candidates provided by a bilingual dictionary are equally likely

to be selected. The regularizer approach has been widely used in many well-known machine learning models, including the logistic regression model [117] and support vector machines [29]. Parameter C_p is introduced to balance the trade-off between the overall coherence measurement and the regularizer. Another important issue with the objective function in (5.22) is the choice of similarity measurement $\mathbf{O}$. A different similarity matrix $\mathbf{O}$ can result in rather different performance in information retrieval. We will show two of them in the later sections. Finally, we would like to emphasize that by solving the resulting optimization problem in (5.22), we are able to acquire the translation probabilities of *all* query words simultaneously through the computation of matrix $\mathbf{P}$. This is in contrast to a number of existing approaches for dictionary-based CLIR, where the selection of translations for individual query words are determined either independently or by certain greedy means that usually leads to suboptimal solutions.

5.3.5 Solving the Optimization Problem

The optimization problem in (5.22) is in fact a quadratic programming (QP) problem [62], since we find the objective function consists exclusively of quadratic terms with respect to the set of variables $\{p_{k,j}\}$ and all the constraints are linear to those variables. A standard QP problem has the following form

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x} + \mathbf{c}^\top \mathbf{x} \\ s.t. \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{E}\mathbf{x} = \mathbf{d} \end{aligned}$$

where the vector $\mathbf{x}$ is the unknown variable, and matrices $\mathbf{H}$, $\mathbf{A}$, $\mathbf{E}$ and vectors $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$ are known. In this section we will discuss how to reformulate our optimization problem (5.22) into the standard QP form, so that it can be easily recognized by most of existing QP solver softwares.

Since the standard QP form takes a vector form of the optimization variables, we first need to rearrange the unknown variables in the matrix $\mathbf{P}$ (i.e. the set of translation probabilities) as a vector; then we need to reformulate the objective function as well as all the constraints in terms of the vector representation of the set of translation probabilities.

To begin with, we concatenate all the row vectors in the matrix $\mathbf{P}$ together to form a vector, i.e.,

$$\mathbf{P}_{m^s \times m^t} = \begin{pmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \vdots \\ \mathbf{p}_{m^s}^\top \end{pmatrix} \rightarrow \tilde{\mathbf{p}}_{m^s m^t \times 1} = \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_{m^s} \end{pmatrix} \tag{5.23}$$

where $\tilde{\mathbf{p}}$ is the rearranged vector, consisting exactly the same set of variables in the matrix $\mathbf{P}$.

Then, we need to reformulate the objective function in (5.22) with respect to the vector $\tilde{\mathbf{p}}$, i.e. finding a matrix $\mathbf{H}$ such that $\mathbf{e}^\top \mathbf{P}\mathbf{O}\mathbf{P}^\top \mathbf{e} - C_p \mathbf{e}^\top \mathbf{P}\mathbf{P}^\top \mathbf{e} = \tilde{\mathbf{p}}^\top \mathbf{H} \tilde{\mathbf{p}}$ is satisfied. It is easy to verify that such a matrix $\mathbf{H}$ can be written in a succinct form by using the *kronecker product* operator $\otimes$ [1] as follows

$$\mathbf{H}_{m^s m^t \times m^s m^t} = \mathbf{1}_{m^s \times m^s} \otimes (\mathbf{O} - C_p \mathbf{I}) \tag{5.24}$$

Here $\mathbf{1}$ stands for a matrix of all elements 1.

Similar procedures can be applied to reformulate the constraints in (5.22), with the use the following matrices

$$\mathbf{T}_{m^s \times m^t} = \begin{pmatrix} \mathbf{t}_1^\top \\ \mathbf{t}_2^\top \\ \vdots \\ \mathbf{t}_{m^s}^\top \end{pmatrix}, \qquad \overline{\mathbf{p}}_{m^s m^t \times 1} = \begin{pmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \\ \vdots \\ \mathbf{t}_{m^s} \end{pmatrix} \tag{5.25}$$

$$\mathbf{E}_{m^s \times m^s m^t} = diag(\mathbf{t}_1^\top, \mathbf{t}_2^\top, \ldots, \mathbf{t}_{m^s}^\top) \tag{5.26}$$

[1] *Kronecker product* is also known as *Matrix Direct Product*. Given an $m \times n$ matrix $\mathbf{A}$ and a $p \times q$ matrix $\mathbf{B}$, their kronecker product $\mathbf{A} \otimes \mathbf{B}$ is an $mp \times nq$ matrix $\mathbf{C}$ whose (i,j)-th submatrix is $a_{i,j}\mathbf{B}$, $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

And finally the optimization problem in (5.22) can be rewritten in a standard form of the QP problem as follows

$$\max_{\tilde{\mathbf{p}}} \tilde{\mathbf{p}}^\top \mathbf{H}\tilde{\mathbf{p}} \tag{5.27}$$

$$s.t. \quad \mathbf{E}\tilde{\mathbf{p}} = \mathbf{e} \tag{5.28}$$

$$0 \leq \tilde{\mathbf{p}} \leq \overline{\mathbf{p}} \tag{5.29}$$

where $\mathbf{e}$ is a vector with all elements equal to 1, and the matrices $\mathbf{H}, \mathbf{E}$ and the vector $\bar{\mathbf{p}}$ are given in (5.24) - (5.26). In our experiments, the QP package from Matlab [143] is used to solve the above problem.

Remark: For the QP problem formulated in (5.27), the problem size appears to be large because the number of variables in vector $\mathbf{q}$ is $m^s m^t$, i.e., the product between the number of unique query words and the number of distinct translation words provided by the dictionary. However, notice that in the constraint (5.29), $\overline{\mathbf{p}}$, i.e., the upper bound of translation probabilities, is a concatenation of translation vectors $\mathbf{t}_i$ obtained from $\mathbf{T}$, the matrix notation of the bilingual dictionary. Given that most query words only have a few translations, most of the elements in the matrix $\mathbf{T}$ will be zeros. As a result, most elements in the upper bound vector $\overline{\mathbf{q}}$ are zeros, which leads to the zero values for the corresponding translation probabilities in $\mathbf{q}$. Hence, the number of non-zero translation probabilities in $\mathbf{q}$ is no more than the total number of translations provided by the bilingual dictionary for the query words, which is usually much smaller than the product $m^s m^t$. Thus, the computation cost of the Maximum Coherence Model is modest for real CLIR practice, if not overestimated.

5.3.6 Summary and Discussion

By putting together the uncertainty modeling, translation probability learning and the retrieval model, we summarize the steps of applying the proposed framework to

Step 0 Prepare a bilingual dictionary

Step 1 Compute the pair-wise similarity of all the words in the target language that appear in the dictionary as translations

Step 2 Prepare language models for both the source language and the target language

Step 3 When a query in the source language comes
1. Identify the candidate set of translations by the dictionary
2. Extract the similarity information related to the candidate set
3. Compute translation probabilities by solving the QP problem (5.27)
4. Retrieve documents according to the model in (5.19), by feeding in the set of translation probabilities and both language models

Figure 5.1. Steps of applying the proposed framework to Cross Lingual Information Retrieval

CLIR in Figure 5.1. In those steps, pair-wise similarity computation and language model preparation can all be done offline. In query time, most computation comes from solving the QP problem, in addition to the conventional retrieval process.

It is important to note that although in this study we limit ourselves to the bag of words approach without exploring other linguistic structures such as phrases, our framework can essentially be generalized to the linguistic structures other than the bag of words. This can be achieved by treating all the w_j^t and w_k^s as the units in the targeted linguistic structures, and $\Pr(w_j^t|w_k^s)$ as the likelihood of translation between the units in the new linguistic structures. We focus our discussion on word translation mainly because of the following two concerns:

1. As mentioned in the introduction part, one of the main motivation behind our work is to resolve the problem of CLIR when only bilingual dictionaries are available. Since in practice most bilingual dictionaries are word-based, we focus our study on the word-based approaches.

2. Since many user queries, particularly the ones from the World Wide Web, are less likely to be grammatically structured, we believe it is important for a robust

CLIR framework to minimize its dependence on deep linguistic analysis of user queries.

Despite of the above claim, we believe that given more linguistic resources available and well grammatically structured queries, our framework could be improve by appropriately incorporating the linguistic constraints derived for the translated queries. For example, we can incorporate the linguistic knowledge into the similarity matrix **O**; or we can introduce it into the regularizer in the optimization problem (5.22) to make it more informative. These framework extensions will be considered in future work.

5.4 Maximum Coherence Model

In the previous studies of selection-based approaches, several metrics have been used for measuring the similarity between two words in the target language, i.e. the $o^t_{j,j'}$, including variants of mutual information [57, 81], and the Dice similarity [1, 2]. In this model, a typical variant of mutual information (which has been used in previous studies [58]) is used as the pair-wise similarity metric

$$s^t_{j,j'} = \Pr(w^t_j, w^t_{j'}) \times \log \frac{\Pr(w^t_j, w^t_{j'})}{\Pr(w^t_j) \times \Pr(w^t_{j'})} \tag{5.30}$$

$\Pr(w^t_j)$ is the unigram probability for word w^t_j, and $\Pr(w^t_j, w^t_{j'})$ is the joint probability for word w^t_j and $w^t_{j'}$ to co-occur in the same documents. Both probabilities can be acquired by simply counting the term frequency of single words and the frequencies of co-occurrence between two words. Note that Equation (5.30) is different from the standard definition for mutual information in that only co-occurrence information is used. Due to the computation concern, in Equation (5.30) the correlation between two words when at least one of them does not occur in documents is ignored. According to the definition in Equation (5.30), we see that two words will be regarded as similar if they co-occur often in the document collection.

Using matrix notation $\mathbf{S} = [s^t_{j,j'}]_{m^t \times m^t}$ and substitute the coherence matrix $\mathbf{O}$ with the similarity matrix $\mathbf{S}$, we have the following model

$$\begin{aligned} \max_{\mathbf{P} \in \mathbb{R}^{m_s \times m_t}} \quad & \mathbf{e}^\top \mathbf{P}\mathbf{S}\mathbf{P}^\top \mathbf{e} - C_p \mathbf{e}^\top \mathbf{P}\mathbf{P}^\top \mathbf{e} \\ s.t. \quad & \mathbf{P} \cdot \mathbf{e}_{m^t \times 1} = \mathbf{e}_{m^s \times 1} \\ & \mathbf{0} \leq \mathbf{P} \leq \mathbf{T} \end{aligned} \tag{5.31}$$

We call the above model "Maximum Coherence Model" since it is a straightforward implementation of the proposed framework. Parameter C_p can be determined empirically by cross validation. Heuristically, we hope C_p to be roughly in the same scale as those elements in the matrix $\mathbf{S}$, since C_p is used to balance between the similarity measurement $\mathbf{S}$ and regularizer $\mathbf{I}$. Hence, we heuristically propose to set C_p to be proportional to the average of the elements in $\mathbf{S}$, i.e.

$$C_p = \eta \sum_{j,j'} s^t_{j,j'} / (m^t)^2$$

Here η is a scaling factor. In our experiments we tested different values of η in the range of $[0.1, 10]$ and chose the one with good retrieval performance.

5.5 Spectral Query Translation Model

One problem with the Maximum Coherence Model proposed in the previous section is the difficulty in determining the value of parameter C_p. This problem arises because the two terms in objective function in (5.31), namely the overall coherence measurement of translated query, and the regularizer, are in different scales. As a result, we have to search for appropriate C_p empirically. In order to put the two terms on the same scale, we introduce the concept of normalized similarity matrix: for a given similarity matrix $\mathbf{S} = [s^t_{j,j'}]_{m^t \times m^t}$, we define a diagonal matrix $\mathbf{D} = diag(d_1, d_2, \cdots, d_n)$, where $d_j = \sum_{j'=1}^{m^t} s_{j,j'}$. Then, the normalized similarity

matrix is $\bar{\mathbf{S}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$. Note that through this normalization procedure, we are able to bring down the scale of matrix $\mathbf{S}$ to $O(1)$. As a result, both the coherence term and the regularization term are on the same scale. Thus, instead of finding appropriate C_p empirically, we can simply set it to 1, which leads to the following realization of the proposed framework:

$$\begin{aligned} \max_{\mathbf{P}\in\mathbb{R}^{m_s\times m_t}} \quad & \mathbf{e}^\top\mathbf{P}\mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}\mathbf{P}^\top\mathbf{e} - \mathbf{e}^\top\mathbf{P}\mathbf{P}^\top\mathbf{e} \\ s.t. \quad & \mathbf{P}\cdot\mathbf{e}_{m^t\times 1} = \mathbf{e}_{m^s\times 1} \\ & \mathbf{0} \le \mathbf{P} \le \mathbf{T} \end{aligned} \tag{5.32}$$

We call the above model "Spectral Query Translation Model" because it can be interpreted as a graph partitioning approach for query translation disambiguation, which is strongly related to spectral clustering. We will further elaborate on this interpretation in the following subsection.

5.5.1 Query Translation Disambiguation as Graph Partitioning

In order to see the relationship between graph partitioning and dictionary-based CLIR model in (5.32), for a given query $\mathbf{q}^s$ and its translation candidate set $\mathbf{R}$, we present the related similarity information $\mathbf{S}$ through an undirected weighted graph. Each translation candidate $w_k^t \in \mathbf{R}$ is represented by a vertex. Any two translation candidates related to two different query words are connected by an edge if they ever co-occur in at least one document. A non-negative weight is assigned to each edge to indicate the similarity between the two connected words. Here we use the measurement defined in Equation (5.30) as the edge weight.

With the constructed graph for a given query and its translation, we hypothesize that the best (or the most coherent) translation of a query corresponds to the most strongly connected component within the graph. To separate the strongly connected component from the rest of the graph, a graph partitioning algorithm can be employed

to divide the graph into two disjoint clusters: a cluster for strongly connected component and a cluster for the rest of the graph. To this end, we inherit the idea from the Normalized Cut algorithm. For the graph constructed for the translation candidate set $\mathbf{R}$, its adjacency matrix is exactly the similarity matrix $\mathbf{S} = [s^t_{j,j'}]_{m^t \times m^t}$, and the graph Laplacian matrix is $\mathbf{L} = \mathbf{D} - \mathbf{S}$. Following the formalization of the Normalized Cut algorithm [134], the optimal 2-way partitioning is found by solving the following minimization problem

$$\min_{\mathbf{v}} \mathbf{v}^\top \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} \mathbf{v} \tag{5.33}$$

Here $\mathbf{v} = [v_1 v_2 \cdots v_{m^t}]^\top$ is a cluster membership indicator vector. Each element v_i is a binary variable with 1 indicating the corresponding word being included in the query translation and 0 for not being included.

It is not difficult to see that the optimization problems in (5.32) and (5.33) are in fact equivalent if we set

$$\mathbf{v} = \mathbf{P}^\top \mathbf{e} \tag{5.34}$$

or in another more explicit form

$$\begin{aligned} v_j &= \sum_k p_{k,j} \\ &= \sum_{w^s_k \in \mathbf{q}^s} \Pr(w^t_j | w^s_k, \mathbf{q}^s) \Pr(w^s_k | \mathbf{q}^s) \cdot m^s \\ &= m^s \cdot \Pr(w^t_j | \mathbf{q}^s) \end{aligned} \tag{5.35}$$

where $\Pr(w^s_k|\mathbf{q}^s)$ takes a uniform distribution, i.e. $\Pr(w^s_k|\mathbf{q}^s) = 1/m^s$.

What is suggested by Equation (5.34) or (5.35) is to relax the cluster indicator v_j to a soft membership instead of a binary value. This soft membership, from the graph partitioning point of view, indicates how likely the strongly connected component will include the particular translation candidate w^t_j. Equation (5.35) links the soft membership with the probability $\Pr(w^t_j|\mathbf{q}^s)$, i.e. the likelihood of including

translation candidate w_j^t in the translation of query $\mathbf{q}^s$. Thus, the model proposed as the optimization problem (5.32) can be perfectly explained from a graph partitioning perspective. Note that the relaxation of a binary indicator to a soft membership in the perspective of graph partitioning is in accordance with our introduction of translation probabilities in the statistical framework. This is because both of them try to address the uncertainty problem in the process of query translation.

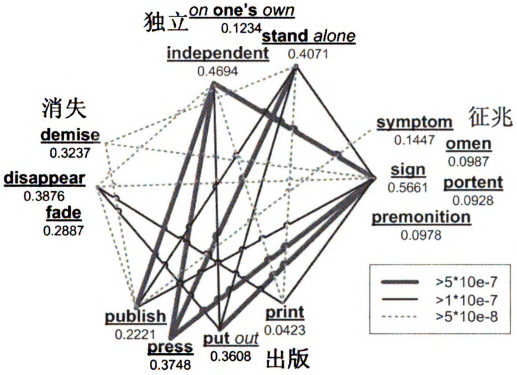


Figure 5.2. An example of graph partitioning perspective for query translation disambiguation.

Figure 5.2 gives an illustrative example of this graph partitioning perspective. In the example, the query is composed of four Chinese words, and around each Chinese word are its translation candidates in English provided by a Chinese-English dictionary. The thickness of lines connecting two English words roughly represents their similarity. The number below each English word is its translation probability estimated from the Spectral Query Translation Model. Based on the graph repre-

sentation in Figure 5.2, we can easily see a strongly connected component consisting of words "independent", "sign", and "press", which have been assigned with large translation probabilities. It is important to note that Figure 5.2 only serves as an illustration of the proposed spectral query translation model. In particular, *all* translation candidates will be used in the retrieval model (see Section 5.3.3), and their importance will be weighted by their translation probabilities determined by the optimization algorithm.

Remark: In the above, we discuss the relationship between the Spectral Query Translation Model and the spectral graph partitioning. However, it is worth pointing out that the solution to the Spectral Query Translation model can not be acquired by the eigen analysis that is used for solving spectral graph partitioning. This is because the Spectral Query Translation model seeks for the optimal translation probabilities (which leads to soft cluster memberships) that maximize the overall coherence measurement. In contrast, most spectral graph partitioning algorithms, such as Normalized Cut, search for the binary cluster memberships that minimize the graph cut. It is such difference that leads to the quadratic programming formalization, instead of an eigenvector problem, for the Spectral Query Translation Model.

5.5.2 Maximum Coherence vs. Spectral Graph Translation

Aside from its graph partitioning explanation, the Spectral Query Translation Model is advantageous to the Maximum Coherence Model in that it is able to reduce the translation probabilities for the "common" words, which may otherwise dominate in the final query translation. To see this, consider an element $\bar{s}_{j,j'}$ in the normalized similarity matrix $\bar{\mathbf{S}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{S}\mathbf{D}^{-\frac{1}{2}}$

$$\bar{s}^t_{j,j'} = \frac{s^t_{j,j'}}{\sqrt{\sum_{j'=1}^{m^t} s^t_{j,j'}}\sqrt{\sum_{j'=1}^{m^t} s^t_{j,j'}}} \tag{5.36}$$

Suppose translation candidate w_j^t is a common word that appears frequently in the document collection of the target language. This implies that the sum $\sum_{j'=1}^{m^t} s_{j,j'}^t$ will be large. Since most of the "common" words are less informative for the purpose of document retrieval, we use the normalization factor $1/\sqrt{\sum_{j'=1}^{m^t} s_{j,j'}^t}$ to suppress their coherence values, which will eventually reduce the probability of including common words in the final query translation.

Based on the above analysis, we see that the Spectral Query Translation Model is more appealing than the Maximum Coherence Model. This is further confirmed by our empirical studies on cross-language information retrieval presented in the next section.

5.6 Experiments and Discussions

The goal of this experiment is to examine the effectiveness of the proposed statistical framework for cross-language information retrieval. In particular, four research questions will be addressed in this empirical study:

1. *Is the proposed statistical framework effective for cross-language information retrieval?* To obtain a comprehensive view, we compare the Maximum Coherence Model and the Spectral Query Translation Model to the existing selection-based approaches using a variety of queries and documents.

2. *How important is it for a query disambiguation algorithm to include translation uncertainty in its analysis?* To address this question, we will examine the importance of including translation uncertainty in cross-language information retrieval through case studies.

3. *How important is it to remove the translation independence assumption for cross-language information retrieval?* To address this question, we will exam-

ine the impact of the translation independence assumption on cross-language information retrieval through case studies.

4. *How does the Maximum Coherence Model empirically compare to the Spectral Query Translation Model?* We will show a performance comparison between the two models as well as a case study as empirical evidence to our previous comparative analysis of the two models.

5.6.1 Experiment Setup

All our experiments are retrieval of English documents using Chinese queries. The document collections used in this experiment are from TREC ad hoc test collections, including

AP88-89 164,835 documents from Associated Press(1988, 1989)

WSJ87-88 83,480 documents from Wall Street Journal (1987, 1988)

DOE1-2 226,087 documents from Department of Energy abstracts [2]

In addition to the homogeneous collections listed above, we also test the proposed model against heterogeneous collections that are formed by combining multiple homogeneous collections. In particular, two heterogeneous collections are created: collection AP88-89 + WSJ87-88, and collection AP89 + WSJ87-88 + DOE1-2. In a heterogeneous collection, words are more likely to carry multiple senses than words from a homogeneous collection, which will increase the difficulty for an automatic algorithm to disambiguate the senses of query words using the pairwise word similarities. The SMART system [126] is used to process document collections. Each document is first parsed into tokens with stop words removed, and then tokens are

[2]DOE1-2 collection is not used as one of the homogeneous datasets in our experiments because DOE1-2 collection provides no relevant documents for a majority of the queries used in this experiment. It is only used to create heterogeneous collections by combining with the other two homogeneous collections.

stemmed using the Porter algorithm. Finally, each document is represented as a bag of stemmed words. We also implemented pivoted document length normalization weighting scheme [135] in SMART system for the retrieval process. Since our goal is to illustrate the advantage of the proposed statistical framework, we do not apply more sophisticated procedures for text analysis in our experiment, such as phrase identification.

Our queries come from a manual Chinese translation of TREC-3 ad hoc topics (topic 151-200). To fully examine the effectiveness of the proposed models, we test it against both the long Chinese queries and the short Chinese queries. A short Chinese query is created by translating the "title" field of an English query into Chinese; a long Chinese query is formed by combining the Chinese translations of both the "title" field and the "description" field in an English query. The average length of short Chinese queries is 9.64 Chinese characters, and 30.72 Chinese characters for long queries. For Chinese translations, we also manually segment the sentences into words with stop words removed, then feed them into our query translation framework. Figure 5.3 gives an example of the title field and description field (in the bottom panel), which is used to form a Chinese query in our experiments.

Since most of the words in a short query are highly relevant to the topic of the query, we would expect that query disambiguation approaches based on word similarities will work well. The analysis of CLIR with long queries could be tricky because of the following two conflicting aspects: (1) On one hand, a long query provides significantly richer context than the short one in disambiguating the word sense of translation. Therefore, we would expect that the long queries may achieve a better performance than the short queries in CLIR. (2) On the other hand, a long query tends to include words either irrelevant or only slightly relevant to its topic. As a result, even a translation word that is coherent with the translations of many query words may not necessarily be a good candidate for selection. In our experiment, we

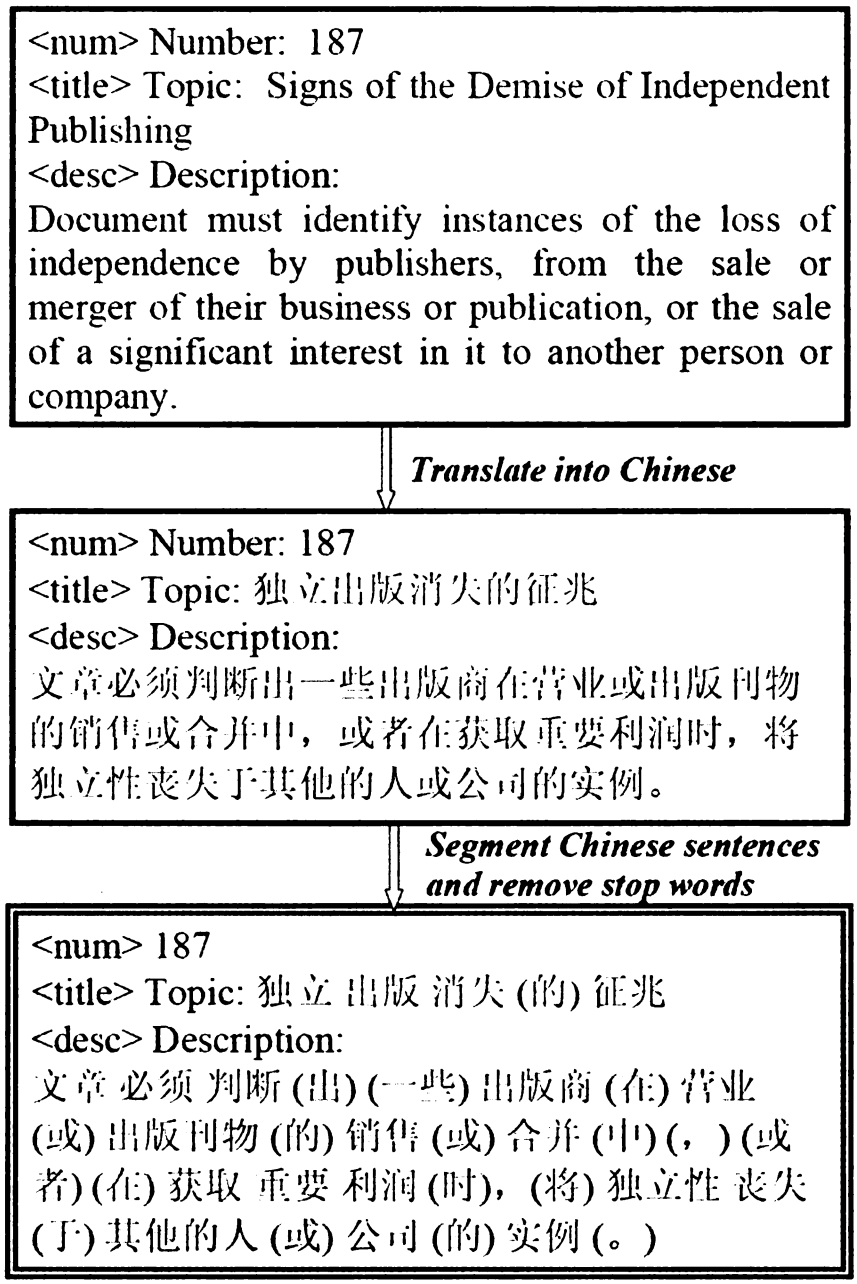


Figure 5.3. An example query used in our experiments. The query is first translated from the No. 187 in TREC-3 ad hoc topics into Chinese. It is then segmented into words with stop words removed. The upper panel shows the original query in English; the middle panel shows the translated Chinese query; the bottom panel shows the segmented Chinese query, where stop words in parentheses are removed.

will examine which factor among the two plays the major role in CLIR. Hence, a long query may pose a more challenging problem than a short query for a translation disambiguation algorithm based on word similarity information.

Finally, the relevance judgments for the original English queries are used as the relevance judgments for their Chinese translations. The Chinese-English dictionary used in our experiments comes from Linguistic Data Consortium (LDC, http://www.-ldc.upenn.edu), which consists of translations for 53061 Chinese words. Since our experiments do not involve the processing of English phrases, for any English phrase that is the translation of a Chinese word, we simply treat it as a bag of words.

To evaluate the effectiveness of the proposed framework and models, we implement two baseline models that take translation selection approaches. The first baseline model selects the most likely translation for each query word, which we call "BSTO". Specifically, we follow the "Approximate Translation Selection Algorithm" described in Section 5.2.1. The second model, which we call "ALTR", makes no efforts for translation disambiguation by simply including all the translations provided by the dictionary for query words into the final query translation. Finally, for easy reference, we use the abbreviation "MAC" for our Maximum Coherence Model and "SQT" for our Spectral Query Translation Model. The constant C_p for the regularizer term in the Maximum Coherence Model is set to be $4\sum_{j,j'} s^t_{j,j'}/(m^t)^2$ based on our empirical experience.

5.6.2 Comparison to Selection-based Approaches

Table 5.1 lists the average precision across 11 recall points for both the homogeneous collections and the heterogeneous collections. As indicated in Table 5.1, the proposed models (i.e., "MAC" and "SQT") outperform the two baseline models for both short queries and long queries across all four different collections. For the purpose of reference, we also list the results of monolingual information retrieval in the first column of

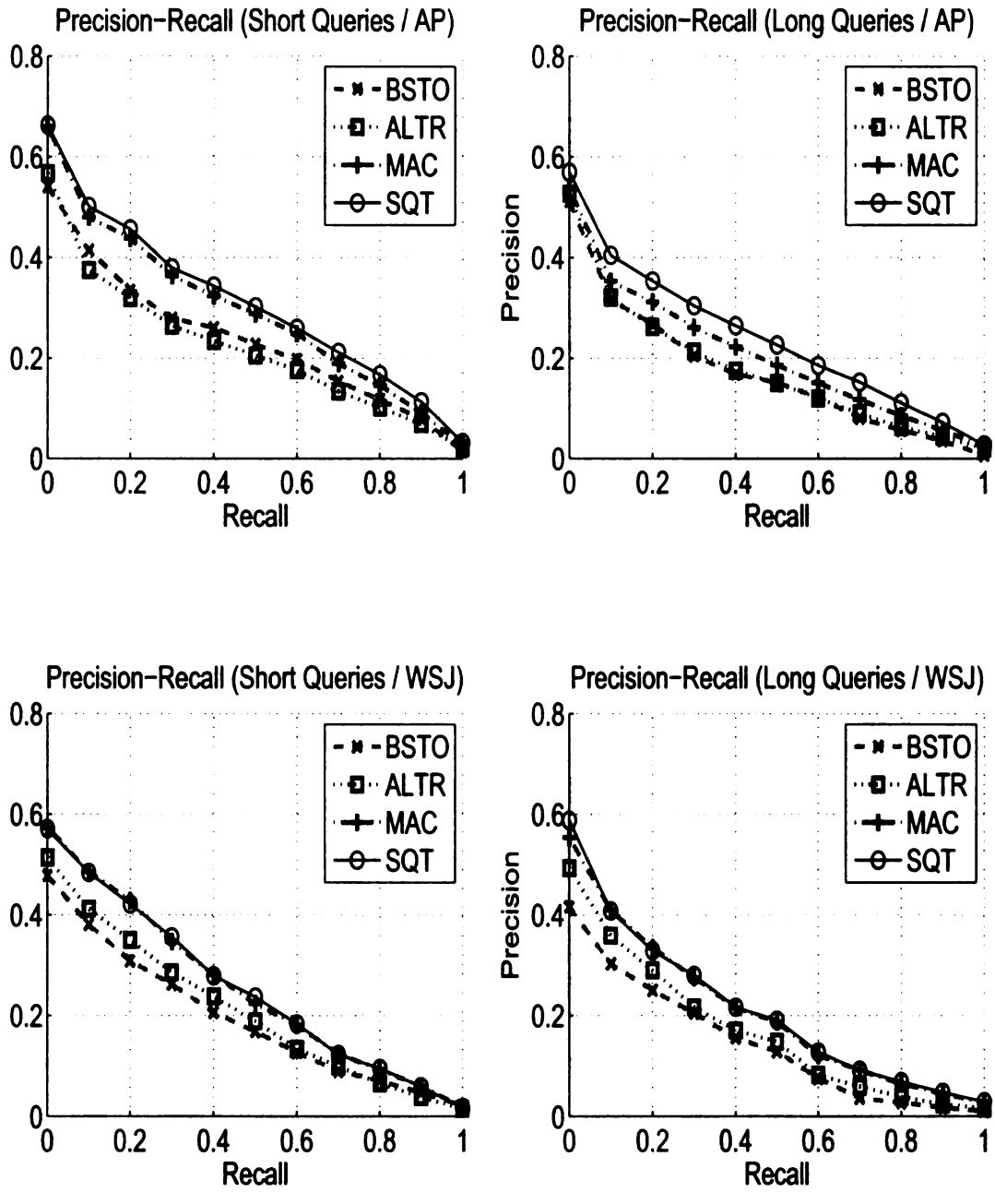


Figure 5.4. Comparison of CLIR performance on homogeneous datasets using both short and long queries. The upper two figures are for AP88-89 dataset, and the lower two are for WSJ87-88 dataset. The left two figures are for short queries, and the right two are for long queries.

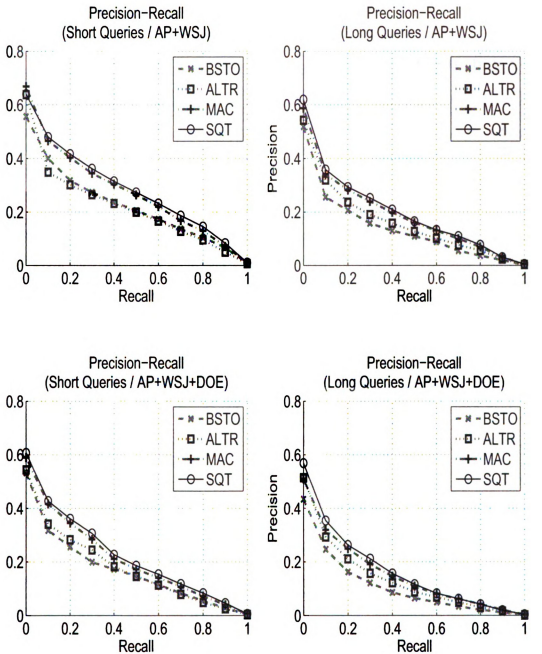


Figure 5.5. Comparison of CLIR performance on heterogeneous datasets using both short and long queries. The upper two figures are for AP88-89 + WSJ87-88, and the lower two are for AP89 + WSJ87-88 + DOE1-2 dataset. The left two figures are for short queries, and the right two are for long queries.

Table 5.1. 11-point average precision for both short and long queries on TREC datasets. MLIR represents the monolingual information retrieval. The relative CLIR improvements of MAC and SQT models over the other two baseline models are listed in the 5th, 6th, 8th and 9th data columns.

	MLIR	BSTO	ALTR	MAC	(M-B)/B	(M-A)/A	SQT	(S-B)/B	(S-A)/A
Short Queries									
AP	.4112	.2381	.2241	.2959	+24.28%	+32.04%	.3116	+30.87%	+39.05%
WSJ	.3386	.1966	.2129	.2560	+30.21%	+20.24%	.2571	+30.77%	+20.76%
AP+WSJ	.4067	.2253	.2209	.2772	+23.04%	+25.49%	.2859	+26.90%	+29.43%
AP+WSJ+DOE	.3556	.1739	.1829	.2172	+24.90%	+18.75%	.2296	+32.03%	+25.53%
Long Queries									
AP	.3756	.1749	.1803	.2096	+19.84%	+16.25%	.2426	+38.71%	+34.55%
WSJ	.4022	.1478	.1727	.2116	+43.17%	+22.52%	.2161	+46.21%	+25.13%
AP+WSJ	.3721	.1433	.1665	.1947	+35.87%	+16.94%	.2048	+49.92%	+23.00%
AP+WSJ+DOE	.3299	.1122	.1411	.1576	+40.46%	+11.69%	.1712	+52.58%	+21.33%

Table 5.1. Furthermore, we plot the precision-recall curves for both the short queries and the long queries in Figure 5.4 and Figure 5.5, respectively. As illustrated in Figure 5.4 and 5.5, for all four collections the precision-recall curves of the proposed models always stay above the curves of the other two baseline models. Based on these results, we conclude that the proposed statistical framework performs substantially better than the other two selection-based approaches for cross-language information retrieval.

A further examination of results in Table 5.1 gives rise to the following observations:

1. In general, the retrieval accuracy for heterogeneous collections appears to be worse than that for homogeneous collections. In particular, a substantial decrease in the average precision is observed for all four models when the collection of DOE1-2 is included in the heterogeneous collection. This result is in accordance with our previous analysis, i.e., words from heterogeneous collections are more likely to have multiple senses, thus resulting in higher translation ambiguity.

2. A better retrieval is achieved for short queries than for long queries. The degra-

dation in performance between long queries and short queries is more significant for heterogeneous collections than for homogeneous collections. Usually long queries bring rich context as well as noise. Our result indicates that among the two factors, the second one, i.e., long queries consists of many irrelevant words, seems to be more influential than the first one, i.e., long queries provide rich context for word sense disambiguation, in query translation. This result seems to contradict the general belief that long queries tend to yield better retrieval accuracy than the short ones given its rich context. However, it is worth pointing out that this general belief is based on a simplified analysis and overlooks the fact that long queries tend to include irrelevant words that could corrupt the accuracy of query translation disambiguation. To further confirm our hypothesis, we compare the results of the monolingual information retrieval between long queries and short queries. We observe that the short queries in fact outperform the long queries for three among four datasets. All the results indicate that due to the two conflicting factors related to long queries, it is not necessary the case that information retrieval with long queries will definitely deliver better retrieval accuracy than short queries.

3. The "BSTO" model does not consistently outperform the "ALTR" model. In fact, for the long queries, the "ALTR" model performs better than the "BSTO" model across all four different collections. This phenomenon can also be attributed to the fact that long queries are rather noisy and likely to include irrelevant words. This result indicates that the "BSTO" model can be sensitive to the noises present in queries. Given that a significant amount of noise can be present in queries, it is important to maintain the uncertainty of translation in the retrieval process. Note that our results appear to be inconsistent with the finding in [57]. We believe that this difference could be explained by the difference in the experiment setups. In particular, since our experiments focus on

	upheaval	commotion	turbulence	unrest	turmoil
动荡	0.19931	0.19723	0.19882	0.20209	0.20255

	intent	spring	motive	inducement	incentive
动机	0.22821	0.19645	0.30384	0.13196	0.13954

	acid	sour	sore	ache
酸	0.79070	0.06422	0.07552	0.06956

Figure 5.6. Examples of translation probabilities estimated by the Maximum Coherence Model.

CLIR with a bag of words, we did not employ any linguistic tools other than removing stop words and stemming keywords. In contrast, in [57] linguistic tools are used to identify appropriate English phrases and their Chinese translation, which has been shown as an important factor in CLIR [8, 57]. Although phrase analysis is important to CLIR, we believe that a generic probabilistic model is beneficial to CLIR of any languages, particularly when linguistic resources are scarce. Other differences between our baseline "BSTO" model and the model in [57] include the different ways of mutual information computation and slightly different translation selection strategies.

5.6.3 The Necessity of Including Translation Uncertainty

To demonstrate the uncertainty in query translation, in Figure 5.6, we list the translation probabilities for three Chinese words [3] that are estimated by the Maximum Coherence Model. As we can see, a significant variance exists in the distribution of translation probabilities across different Chinese words. The first example in the figure shows an almost uniform distribution over all translations, while the third one illustrates a very skewed distribution. Meanwhile, the second example provides a distribution that is neither uniform nor totally skewed. These three examples illustrate

[3]These three Chinese words are not from the same query.

CH Term	EN Translation	Selection (BSTO)	Trans. Prob. (MAC)	Trans. Prob. (SQT)
独立	independent		0.36208	0.46944
	on one's *own*		0.24481	0.12342
	stand *alone*	x	0.39311	0.40715
出版	press	x	0.33789	0.22208
	publish		0.19973	0.37477
	put *out*		0.33311	0.36082
	print		0.12927	0.04232
消失	disappear	x	0.34941	0.38760
	demise		0.32035	0.32365
	fade		0.33024	0.28875
征兆	symptom		0.16698	0.14468
	omen		0.16155	0.09868
	sign	x	0.35016	0.56612
	portent		0.16097	0.09279
	premonition		0.16034	0.09773

Original TREC Topic in English (topic 187 'title' field):
Signs of the Demise of Independent Publishing

Figure 5.7. An example of query translation, using the "BSTO" model and the Maximum Coherence Model. (English words in italicized font are removed as stop words.)

the "*translation uncertainty problem*", which we have addressed in previous sections. Furthermore, the diversity in the distribution of translation probabilities makes it difficult for a selection-based approach to perform well over all different cases. For example, the "BSTO" model is able to work well for the third example but will fail in the first one. On the other hand, the "ALTR" model would be perfect for the first example but not for the third one. Base on the above analysis, we conclude that it is important to capture the translation uncertainty and the diversity of translation uncertainty in a probabilistic model.

5.6.4 The Impact of Translation Independence Assumption on Query Disambiguation

To illustrate the impact of the translation independence assumption on query translation disambiguation, consider the example in Figure 5.7. This query consists of four Chinese words, and the English translations for each Chinese are provided by the dictionary are listed in the second column. The original English query is also included at the bottom of the figure. The English translations selected by the "BSTO" model are listed in the third column, marked by small crosses. The translation probabilities from Chinese words to their English translations estimated by the Maximum Coherence Model and the Spectral Query Translation Model are listed in the last two columns respectively.

Comparing to the original English query, we see that the "BSTO" model makes incorrect translation selection for both the first and the second Chinese words. For the first one, the correct English translation should be "independent", instead of "stand (alone)" [4]. The better translation for the second Chinese word should be "publish" instead of "press". One reason for such mistakes is that in the "BSTO" model, the coherence score of a translation is computed based on all the English translations provided by the dictionary for the Chinese words in the query. Thus, the coherence score of one translation word is completely independent from the selection of other translations. Since both "stand" and "press" are common in English, their overall coherence scores turn out to be larger than the coherence scores of other words, which lead them to be selected by the "BSTO" model. In contrast, in both the Maximum Coherence Model and the Spectral Query Translation Model, the estimation of translation probabilities for one word is dependent on the estimation of translation probabilities for other words. As a result, they are able to adjust the mistakes by assigning significant

[4] "alone" is removed as a stop word and does not count in the translation. It is listed only for the sake of clarity.

amounts of probability mass to the correct translations. For example, for the first Chinese word, both models are able to assign a probability to the correct English translation "independent" comparable to the probability assigned to the translation "stand (alone)".

Note that neither the Maximum Coherence Model nor the Spectral Query Translation Model is able to *always* assign the largest probabilities to the best translation candidates (such as for the first Chinese word in Figure 5.7). However, in general by maximizing the coherence of the entire translated query both models tend to shift more probability mass to the best translation candidates, thus alleviate the mistakes brought by those selection-based approaches originated from their false assumption on translation independence. We believe this is one of the major advantages of these two models over all the selection-based approaches.

5.6.5 Performance: MAC vs. SQT

侵蚀	encroach	erode	weather	eat
MAC	0	0.01091	0.34901	0.14008
SQT	0.02389	0.03139	0.34487	0.09985

Original TREC Topic in English (topic 188 'title' field):
Beachfront Erosion

Figure 5.8. Comparison of an example query word translation using Maximum Coherence Model and Spectral Query Translation Model. The numbers showed in this example are the probabilities of the translation candidates being included in the final query translation.

Table 5.1 reveals a slightly higher average precision across 11 recall points of "SQT" model compared to "MAC" model. In Figure 5.4 and Figure 5.5 the precision-recall curves of "SQT" model generally stays above those of "MAC" model, though the margin is not clear sometimes. All the experiment results suggest that the Spectral Query Translation Model is slightly better than the Maximum Coherence Model.

This observation is in accordance with our theoretical analysis on the two models at the end of Section 5.5. Apart from the advantage of saving the trouble of choosing the optimal regularizer constant C_p, the benefit of normalizing the coherence matrix can be observed from the example presented in Figure 5.8, where we list and compare the probabilities of including different translation candidates in the final query translation for one example query word from both models. As we can see, the common word "eat"is assigned with a significant amount of probability mass in the Maximum Coherence Model although it is almost irrelevant to the context of the entire query. At the same time, the word "encroach", which in fact is related to the theme of the query, receives an almost zero probability (or too small to represent). As indicated by the results listed in Figure 5.8, the Spectral Query Translation Model is able to overcome this problem by re-distributing some of the probability mass on the common word "eat" to other words. This is consistent with our previous analysis that the Spectral Query Translation Model has a better way to estimate the translation probabilities of common words than the Maximum Coherence Model.

5.6.6 Computational Efficiency

To examine the computational efficiency, we calculate the averaged number of seconds that are spent by the proposed algorithms to solve the optimization problem for each query. All the experiments are conducted on a PC with a Pentium 4 2.8GHz CPU and 1G RAM that runs Matlab 7.1 on a Windows system. We directly use the quadratic programming function provided by Matlab to solve the QP problem in the proposed algorithms for query translation disambiguation. The results are 0.027 seconds per query for short queries, and 3.014 seconds per query for long queries. Clearly, our algorithm is sufficiently fast for short queries, but a little slow for long queries. To improve the computational efficiency of long queries, we can first divide a long query into a number short ones, and then translate each short query using the

proposed framework. Since the computational complexity of quadratic programming is $O(n^3)$ in the worst case where n is the number of translation probabilities, by significantly reducing the number of query words, we reduce the number of translation probabilities, and therefore improve the computational efficiency. Note that the above approach is based on the assumption that most of the context needed for query translation disambiguation can be found in the neighborhood of each query word. As a potential research issue, improving the computational efficiency for long queries is worth further study.

5.7 Conclusions

In this chapter, we propose a novel statistical framework for cross-language information retrieval. It utilizes word co-occurrence statistics for estimating translation probabilities that are effective for query disambiguation. Compared to the selection-based approaches, the merits of the framework are twofold: 1) It preserves the translation uncertainty through the estimation of translation probabilities; 2) It estimates the translations for all query words simultaneously. Two realizations of the proposed framework, namely the Maximum Coherence Model and the Spectral Query Translation Model, are presented based on different choices of coherence measurements. Our analysis indicates that the Spectral Query Translation model can also be interpreted as a graph partitioning approach for query translation disambiguation. Empirical results under various scenarios have shown that the proposed framework is able to perform substantially better than the existing selection-based approaches. Further analysis indicates that the Spectral Query Translation Model is more effective and reliable than the Maximum Coherence Model when dealing with common words.

CHAPTER 6

Semi-supervised Learning for Extraction of Questions and Answers from Web FAQs

6.1 Introduction

Question-Answering is faced by a problem of a "lexical gap" [22] or a "word mismatch" [156] between question and answers, suggesting that retrieval of candidate documents for answer extraction should focus on documents that are similar to the expected answer rather than on documents that are similar to the user question. Recent work in Question-Answering has capitalized on existing repositories Frequently-Asked-Question (FAQ) pages to bridge this lexical gap. FAQ pages are easily available in large quantities on the web. They cover a wide range of topics, and thus present an ideal resource to learn about question-answer correspondences. Large repositories of question-answer (QA) pairs extracted from FAQ pages have been used to train statistical translations models in order to provide an additional measure to rank candidate answers [22, 51, 122, 137], or to perform expansion of questions by answer terms using various query expansion techniques [78, 121, 3, 69]. In other work, QA repositories have been used as candidate collections for answer retrieval [30, 83, 154]. Besides Question-Answering, QA repositories have also been used in other related information retrieval and natural language processing tasks, such as query summa-

rization [23], semantically similar question finding [82] and knowledge representation [153, 154]. Since the performance of these applications depends heavily on the quality of the QA repository, high-precision extraction of QA pairs is crucial in order to provide reliable data for various deployments.

While FAQ pages abound on the web, high-precision extraction of QA pairs is challenging because of the wildly varying markup of questions and answers across FAQ pages. We refer to this challenge as the *Across Page Diversity* challenge. Across FAQ pages on the web, the variants in QA markup range from simple paragraph breaks to special formatting (using italics, boldface, different fonts, font sizes, colors, etc.), various types of indentations (lists, tables, etc.), special prefixes (numbers, `Q.`, `Q:`, `Question:`, etc.), or even sophisticated images. The creative power of web designers is displayed not only in the huge variations of QA markup, but also in the endless variants of "noise text", i.e., headers, navigational text, or other annotations that display neither questions nor answers.

To illustrate the *Across Page Diversity* challenge, Figure 6.1 - 6.4 present four snapshots of FAQ pages, each displaying questions and answers in their own format. Specifically, Figure 6.1 presents two FAQ answers in sophisticated formats — one with embedded listing structures and varying fonts, and the other with multiple paragraphs separated by images; the other three snapshots present QA text mostly in plain paragraphs. In Figure 6.2 two types of images prefix each FAQ question and answer; in Figure 6.1 and Figure 6.3 only FAQ questions are marked by numbers; Figure 6.3 uses different colors and fonts to distinguish questions and answers. Figure 6.3 includes a real FAQ question without an ending question mark (i.e. the second question in the snapshot), while Figure 6.4 includes a fake FAQ question that does end with a question mark (i.e., in the dashed ellipse). Also, there are various types of noise texts that are highlighted by boxes of all shapes in the four snapshots.

The diversity of web FAQ pages pose a serious challenge in extracting high-quality

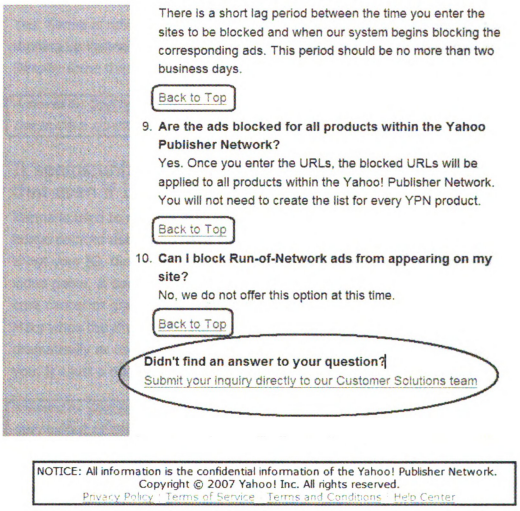


Figure 6.1. Snapshot of example web FAQ pages (1 of 4)

Is there a limit to how much karma you can accumulate?

Yes. Karma is now capped at "Excellent" This was done to keep people from running up insane karma scores, and then being immune from moderation. Despite some theories to the contrary, the karma cap applies to every account.

Answered by: CmdrTaco
Last Modified: 1/24/02

It seems unfair that I can't get any more karma than that even if I earn it.

Karma is used to remove risky users from the moderator pool, and to assign a bonus point to users who have contributed positively to Slashdot in the past. It is not your IQ, dick length/cup size, value as a human being, or a score in a video game. It does not determine your worth as a Slashdot reader. It does not cure cancer or grant you a seat on the secret spaceship that will be traveling to Mars when the Krulls return to destroy the planet in 2012. Karma fluctuates dramatically as users post, moderate, and meta-moderate. Don't let it bother you. It's just a number in the database.

Answered by: CmdrTaco
Last Modified: 10/19/00

Why didn't I get karma for a Quickie or a Slashback story?

This is a shortcoming in the code that we haven't solved yet. Essentially, the

Figure 6.2. Snapshot of example web FAQ pages (2 of 4)

4. How do I uninstall Google Web Accelerator?

If you decide you don't want to use Google Web Accelerator, here's how to uninstall it:

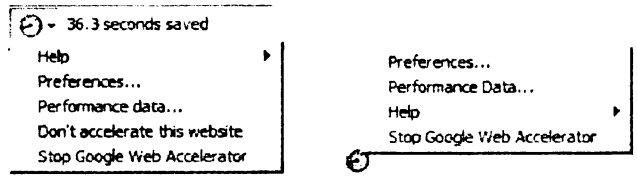
1. Click on **Start > Settings > Control Panel** to open the Windows Control Panel.
2. Click on **Add or Remove Programs** to open its window.
3. Click on **Google Web Accelerator**. A **Remove** button will appear below the Google Web Accelerator item.
4. Click the **Remove** button.
5. Close the Add or Remove Programs window and the Windows Control Panel.

The Google Web Accelerator Menu

[back to top]

1. How do I access the Google Web Accelerator menu?

You can bring up the Google Web Accelerator Menu by clicking on the Google Web Accelerator speedometer icon in the toolbar or system tray.



Learn more about each menu item:

- Preferences
- Performance data
- Stop/Start Google Web Accelerator

Preferences

[back to top]

1. What Google Web Accelerator preferences can I set?

The Preferences menu option opens this Web Accelerator Preferences page:

Figure 6.3. Snapshot of example web FAQ pages (3 of 4)

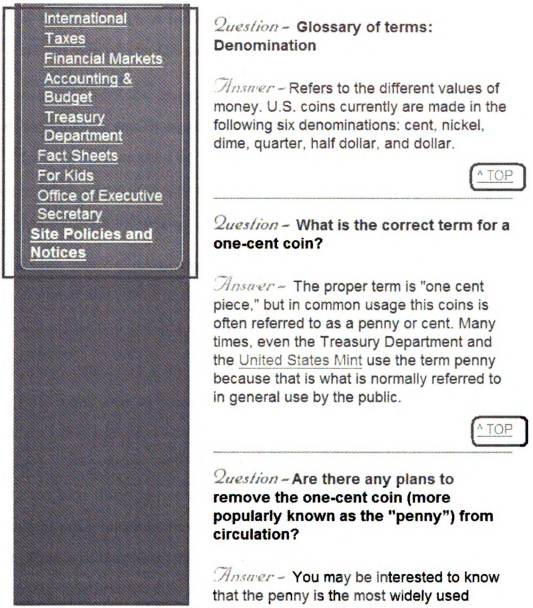


Figure 6.4. Snapshot of example web FAQ pages (4 of 4)

QA from them. At least three major reasons that contribute to the difficulty of the task:

- First, the list of QA pairs within a FAQ page is often mixed with various amount of *noise text*, such as section headings, navigational text, or annotations, that are not part of any QA pair. Separating the text of QA pairs from the noise text can be difficult.

- Second, it is usually significantly more challenging to accurately identify the texts for answers than for questions. This is because the answer text tends to be much longer than that of questions, and more diversified in its presentation format and word usage.

- Third, many questions and answers from the FAQ pages do not follow the grammar and syntax of English rigorously, which makes it difficult to apply the algorithms that are developed in natural language processing.

One may enumerate a few heuristic rules for identifying questions and answers from FAQ pages, such as punctuations, listing markers, lexical cues, repeating patterns, etc., as suggested in several previous studies [136, 84, 137, 99]. However, there are two major problems with employing those heuristics-based approaches. First, due to the large diversity in the presentation formats of questions and answers, it is difficult, if not impossible, to come up with a comprehensive list of heuristics that cover all the possible presentation patterns. Second, many heuristic-based approaches tend to struggle between false positives (i.e., accuracy) and false negatives (i.e, completeness) in QA extraction. More specifically, a set of relaxed heuristic rules tend to find more QA pairs but at the price of including noise text into the QA pairs; by tightening the heuristic rules, we are likely to avoid the problem of including noise text but at the risk of missing many true QA pairs.

Another approach toward QA extraction is to view it as a classification problem of three classes, i.e., the class of questions, answers, and noise. We can then employ the supervised learning algorithm to train a statistical classifier from the labeled examples. However, a key difficulty with this approach is that, due to the large variance in the presentation formats of QAs, it is difficult for any supervised learning algorithm to capture all the possible patterns from a limited number of labeled examples. Apparently, this difficulty originates from the *Across Page Diversity*. Fortunately, while QA markup may vary widely *across* pages, it is generally true that *within* single pages QA markup is consistent. This *Within Page Consistency* principle can be viewed as "side information" – which encodes human knowledge on the fact of FAQ page creation – to the QA extraction task. As will be shown later in this section, the principle's accuracy is well supported by empirical study.

The *Within Page Consistency* leads to the possibility of a bootstrapping approach to QA pair extraction: various heuristics that have been proposed for QA pair extraction can be deployed to perform a high-precision labeling of a seed set of text segments as questions, answers and noise; an optimization problem is solved that essentially propagates the class labels of the seed segments to the text segments that share similar HTML formats as the seeds. This optimization problem can be described in a transductive learning setting, similar to the well-known semi-supervised learning algorithm – Spectral Graph Transducer [87]. Given a large enough database of FAQ pages, which themselves have to be extracted in a high-precision fashion, each of the extraction steps —labeling of seed examples, and similarity-based propagation—can be tuned to high precision, resulting in a large database of correctly identified QA pairs. We show in an experimental evaluation that compared to both heuristics-based and supervised learning approaches, the proposed approach significantly improves precision in QA extraction.

The key advantages of the proposed algorithm are

- It takes full advantage of the heuristics discovered by the previous study of QA extraction.

- It is entirely unsupervised in that all the labeled examples for the semi-supervised learning algorithm are acquired by the heuristic rules. Therefore no human annotation is needed.

- The extraction from one Web FAQ page is independent from any other pages, which enables the algorithm to be carried out in a parallel manner to handle huge amount of data from the Web.

The remaining chapter is organized as follows. In Section 6.2 we briefly review previous work on automatic QA extraction, and the related Spectral Graph Transducer algorithm. In Section 6.4, we present a few important observations on the web FAQs data that motivates our work. The semi-supervised learning algorithm for QA extraction is proposed in Section 6.5. Experiments and discussions are presented in Section 6.6. Finally, Section 6.7 concludes the work.

6.2 Related Work

6.2.1 QA Extraction from the Web

At first glance, extracting QA text from FAQ pages looks like a typical information extraction (IE) task. In [111] a maximum entropy Markov model has been proposed, and in [97] logical structure detection is used. While both approaches prove to be effective in QA text extraction from Usenet FAQ files, they are domain-specific since they rely heavily on the special format of Usenet FAQs. Although IE research have gained significant progress towards open-domain free-text tasks (for example, seminar announcement extraction [36]), the extraction is usually carried out as filling template slots. Unlike seminar announcements that have a relatively fixed set of themes and

presentation formats, QA can be a place-holder for virtually any theme or any presentation format. As a result, it is difficult to apply IE algorithms to extract QA pairs from FAQ pages.

Most of the previous studies on QA extraction from FAQ pages [99, 136, 84, 137] are heuristics-based approaches. In summary, four types of heuristics have proved to be useful for identifying QA text segments in a FAQ page: (1) punctuations, (2) HTML tags (e.g., `<p>`, `<BR>`), (3) listing markers (e.g., `Q:`, `(1)`), and (4) lexical cues (e.g., `What`, `How`). Then rules or memory-based learning algorithms are applied to determine whether or not a text segment is a question or an answer. However, most previous efforts toward building a QA repository only aim at extracting FAQ questions. Only a few studies are related to extracting FAQ answers. The work reported in [99] concentrates on the extraction of FAQ questions, and defines as answers the region between two consecutive FAQ questions, thus ignoring the separation of QA pairs from noise text. In [136, 84, 137], up to three consecutive sentences following each identified question are viewed as the corresponding answer text. As already pointed out in Section 6.1, FAQ answers are significantly more difficult to be extracted than FAQ questions due to their length and diverse presentation formats. An important aspect that distinguishes our work from all the previous studies is that we are aiming at *complete* and *noise-free* text of *both* FAQ questions and answers, which is more challenging yet more useful for most deployments of QA repositories.

6.2.2 Review on Spectral Graph Transducer

Spectral Graph Transducer (SGT) [87] balances between fitting a model that is consistent with supervised information (from labeled examples), and taking the central assumption behind nearly all semi-supervised learning algorithms, i.e., examples are more likely to share the same class labels if they are close to each other or in the underlying manifold. Suppose $\{x_i\}_{i=1}^{n+m}$ is the set of data examples, with the first n

	web pages	FAQ pages	QA pairs
count	4 billion	795,483	10,568,160

Table 6.1. Corpus statistics of QA pair data

examples being labeled and the rest m examples as unlabeled. An $(n+m) \times (n+m)$ matrix $\mathbf{L}$ is defined as the graph Laplacian that can be derived from the pairwise similarities of all the data examples [38, 87]. Note that the graph Laplacian matrix $\mathbf{L}$ captures the structure of both label and unlabeled data. A $(n+m)$-dimension vector $\mathbf{r}$ is used to encodes the label information, where $r_i = \hat{r}_+$ if x_i is a positive example, $r_i = \hat{r}_-$ if x_i is a negative example, and $r_i = 0$ if x_i is unlabeled. $\hat{r}_+$ and $\hat{r}_-$ are constants that depend on the priors of the two classes. Let $\mathbf{f} \in \mathbb{R}^n$ denote the vector of predicted class labels. Spectral Graph Transducer is defined by the following optimization problem

$$\begin{aligned} \min_{\mathbf{f}} \quad & \mathbf{f}^\top \mathbf{L}\mathbf{f} + c(\mathbf{f}-\mathbf{r})^\top \mathbf{C}(\mathbf{f}-\mathbf{r}) \\ s.t. \quad & \mathbf{f}^\top \mathbf{e} = 0 \qquad (\text{where } \mathbf{e} = [1, \cdots, 1]^\top) \\ & \mathbf{f}^\top \mathbf{f} = n+m \end{aligned} \tag{6.1}$$

where the matrix $\mathbf{C} = diag(c_1, c_2, \ldots, c_n)$ is a diagonal cost matrix that assigns a different misclassification cost for each labeled data example. The trade-off between graph cut value (i.e. the first term) and training error (i.e. the second term) is balanced through the constant c.

6.3 FAQ Page Classification

As shown in Table 6.1, the FAQ pages used in our experiment were extracted from a 4 billion page web crawl using the queries "`inurl:faq`" and "`inurl:faqs`" to match the tokens "faq" or "faqs" in the urls. This extraction resulted in 2.6 million web pages (0.07% of the crawl). Since not all those pages are actually FAQs, we manually

labeled 1,000 of those pages to train an online passive-aggressive classifier [41] in a 10-fold cross validation setup. Training was done using 19 features on URLs, question marks and word statistics (see Appendix A.3 for details), and resulted in an F1 score of around 90% for FAQ classification. Application of the classifier to the extracted web pages resulted in a classification of 795,483 pages as FAQ pages. Instead of going into more details of the FAQ page classification algorithm, we will concentrate in this paper on QA pair extraction, and present the main algorithm for QA pair extraction from FAQ pages in following section.

6.4 Observations on Web FAQs Data

To better understand the challenges in extracting QA pairs from the FAQ pages, in this section, we summarize some important observations from the web FAQs data [1] as follows

Noise Text For most FAQ pages, we often find the noise text that do not consist of any questions or answers. These texts can appear either between two consecutive QA pairs or outside the entire list of QA pairs. It is important for the QA extractor to remove the noise text from the identified questions and answers.

Question Mark Rule Most, though not all, FAQ questions end with question marks. This implies that the question mark is an important feature to identify FAQ questions.

Pattern Diversity and Within Page Consistency Due to the heterogeneous authorships of web pages, no specific text formats are consistently used across all the FAQ pages. This implies that it could be very difficult to develop a supervised learning algorithm that is able to capture the large diversity in the

[1]Note that there are two types of FAQ pages: one that compiles multiple QA pairs in a *single* page, and the other that devotes each page to only one QA pair. We will focus on the first type since it is dominant on the web.

presentation patterns of QA pairs across different FAQ pages. However, *within a FAQ page*, it is often the case that consistent HTML formats are used to present questions, answers, and noise text. Moreover, the formats for presenting questions, answers, and others are often different and distinguishable. This motivates us to consider a semi-supervised learning algorithm that explicitly exploits the within page consistency.

The proposed algorithm consists of two major components, i.e., the heuristics that are applied to identify the seed examples of FAQ questions, answers, and noise text, and the semi-supervised learning algorithm that propagate the class labels of the seed examples to other text segments based on the *Within Page Consistency* principle. The following heuristics are used to identify the seed examples:

1. From all the text segments that end with question marks, select the ones whose format repeats the most often and label them as FAQ questions. [2]
2. From all the text segments between any two consecutive FAQ questions that are identified by the first heuristic, select the ones whose format repeats the most often and label them as FAQ answers.
3. If a text segment repeat itself k time in a FAQ page, all those occurrences of the text segment will be labeled as noise text. [3] k is a predefined integer, and is set to 4 in our experiments. And all text segments before the first question are labeled as noise.

All the above heuristics can be seen as a simplified version of those proposed in previous studies that are mentioned in Section 6.2.1. These rules tend to be accurate with very small number of false positives. According to our study, these rules can

[2]Text segments of hyperlinks are excluded, to avoid the possible question lists at the top of the many FAQ pages.

[3]This heuristic is designed to detect those highly repetitive noise text (such as navigational links, e.g., "back to top").

achieve a precision of 96.8% in classification [4].

6.5 A Semi-supervised Learning Approach for QA Extraction

In order to exploit the *Within Page Consistency* principle, in the proposed approach, we view the extraction of QA pairs from each FAQ page as an independent multi-class classification problem, i.e., to classify each text segment in a FAQ page into the classes of questions, answers, and noise text. The key idea behind the proposed algorithm is to first identify a few text segments that can be classified by the specified heuristics. The class labels of these identified segments are then propagated to the text segments of the same web page that are similar in HTML format. In this section, we first review the preprocessing step that divides the FAQ page into text segments, followed by the description of the proposed semi-supervised learning algorithm for QA extraction.

6.5.1 Web Page Preprocessing

The main purpose of preprocessing is to divide a FAQ page into text segments to be classified. In order to exploit the *Within Page Consistency* principle, we need an effective way to represent the presentation format of text segments so that the similarity between any two text segments can be computed accurately. Since all the web FAQs are presented in the HTML file, we propose to divide each FAQ page into text segments based on the layout of the HTML tags, and represent the format of each text segment by the related HTML tags. In particular, we define a *text segment* as a continuous text region that is surrounded by exactly the same pair of immediate

[4]Due to the well-known precision-recall trade-off, when the precision of those rules are tuned to 96.8%, the corresponding recall becomes very low. Therefore these rules, though can achieve high extraction accuracy, are not suitable for *complete* and *noise free* QA extraction

opening and closing tags. For example, given the following HTML text

```
<HTML><HEAD> Title Goes Here </HEAD>
<BODY><H1> This is the heading </H1>
<P> Q1: this is question 1 </P>
<P> A1: answer 1 starts here
<SPAN style="font-size: xx-small;">
  some small font text </SPAN>
<TABLE><TR><TD> table cell 1 </TD></TR>
       <TR><TD> table cell 2 </TD></TR></TABLE></P>
```

the identified text segments are: **"Title Goes Here"**, **"This is the heading"**, **"Q1: this is question 1"**, **"A1: answer 1 starts here"**, **"some small font text"**, **"table cell 1"**, **"table cell 2"**. Given the identified text segments, the next step is to represent the format of each segment by a sequence of HTML tags. In particular, each text segment t_i is represented by a HTML tag sequence $\mathbf{q}_i$ including all the HTML tags that surround segment t_i from outermost to innermost. For example, for the text segment **Q1: this is question 1**, its HTML tag sequence is **HTML, BODY, P**, while the HTML tag sequence for the text segment **some small font text** is **HTML, BODY, P, SPAN**. Finally, given the representation of presentation format, the *format similarity* between two segments t_i and t_j is calculated as follows:

$$sim_f(t_i, t_j) = \exp(-\lambda d(\mathbf{q}_i, \mathbf{q}_j)) \tag{6.2}$$

where $d(\mathbf{q}_i, \mathbf{q}_j)$ is the edit distance between the two corresponding HTML tag sequences $\mathbf{q}_i$ and $\mathbf{q}_j$. λ is a decay factor (set to 0.1 in our experiment).

We find that there is a disadvantage of the web page division scheme we proposed here. In particular, it will break a sentence into multiple text segments if there are hyperlinks or highlighted terms within the sentence. To remedy this problem, we

can detect the broken sentences from the text segments based on a few grammatical rules. For example, a text segment ending with no punctuation and its following text segment starting with an small-cased letter are very likely to be two broken parts from the same sentence. We then adjust the similarity measurement as follows:

$$s_{i,j} = sim_f(t_i, t_j) \cdot b_{i,j} \tag{6.3}$$

where sim_f is the format similarity, and $b_{i,j}$ is an adjustment factor from broken sentence detection, i.e. $b_{i,j}$ takes a constant value $b_0 > 1$ if t_i and t_j are two neighboring text segments that are detected as from the same sentence, and 1 otherwise.

6.5.2 Semi-supervised Learning for QA Extraction

The key idea of the semi-supervised learning approach is to propagate the class labels of the text segments that are classified by the heuristics to the segments of the same page that are similar in the presentation patterns. The idea of label propagation, as shown in [87], is equivalent to minimizing the inconsistency between the similarity of text segments and the class labels assigned to the segments. We can thus encode the idea of label propagation by the following optimization problem:

Let $\{t_i\}_{i=1}^{N}$ denote the set of all N text segments in a FAQ page. We introduce a probability matrix $\mathbf{P} = \left(p_{i,j}\right)_{N \times 3}$, where $p_{i,j}$ indicates the probability of assigning the i-th text segment to the j-th class. Here, we encode the three classes by: class 1 for FAQ questions, class 2 for FAQ answers, and class 3 for noise text. Note that for each segment t_i

$$\sum_j p_{i,j} = 1 \tag{6.4}$$

For convenience, we also use $\mathbf{p}_j$ to denote the j-th column vector in the matrix $\mathbf{P}$, i.e., $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$. Each vector $\mathbf{p}_j$ includes the probability of all the text segments belonging to the corresponding class.

To represent the class information of those seed text segments, we introduce another matrix $\hat{\mathbf{P}} = \left(p_{i,j}\right)_{N\times 3}$, where $\hat{p}_{i,j} = 1$ if the i-th text segment is a seed text segment that is classified into the j-th class by the heuristics, and $\hat{p}_{i,j} = 0$ otherwise. Similarly we can decompose the matrix $\hat{\mathbf{P}}$ into column vectors that represent the seed examples in each class respectively, i.e., $\hat{\mathbf{P}} = [\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \hat{\mathbf{p}}_3]$.

Let a matrix $\mathbf{S} = (s_{i,i'})_{N\times N}$ denote a matrix of all the pairwise similarities of the text segments, as defined in Equation (6.3). We can view the similarity matrix $\mathbf{S}$ as building a weighted graph: each node represents a text segment; if the similarity between two text segments is non-zero, the corresponding two nodes is connected by an edge that is weighted by the similarity value. Based on a weighted graph, we can define a normalized graph Laplacian [38]

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{S}$$

where $\mathbf{D} = \text{diag}(d_1, \cdots, d_N)$ is a diagonal matrix with each $d_i = \sum_{i'} s_{i,i'}$. The graph Laplacian includes the format similarity between any two text segments, and will be used as the basis to exploit the *Within Page Consistency* principle.

Our goal is to estimate the optimal set of probabilities in the matrix $\mathbf{P}$, such that if two text segments are similar in their presentation formats, they will be assigned with similar probabilities to the three classes. In addition, the optimal set of probabilities should also be consistent with the class labels that are assigned to the seed examples by the heuristics. All these can be formulated into the following optimization problem

$$\begin{aligned} \min_{\mathbf{P}} \quad & \sum_{j=1}^{3} \left[\mathbf{p}_j^\top \mathbf{L}\mathbf{p}_j + C_j(\mathbf{p}_j - \hat{\mathbf{p}}_j)^\top \mathbf{\Lambda}_j(\mathbf{p}_j - \hat{\mathbf{p}}_j)\right] \\ s.t. \quad & \sum_{j=1}^{3} \mathbf{p}_j = [1, \cdots, 1]^\top \\ & \mathbf{p}_j \geq \mathbf{0}, \quad j = 1, 2, 3 \end{aligned} \tag{6.5}$$

where $\mathbf{\Lambda}_j$ is a diagonal matrix with a diagonal element equal to 1 if corresponding

text segment is a seed example in the j-th class and 0 otherwise. C_j is a predefined factor that balances the two terms in the objective function.

The first term $\mathbf{p}_j^\top \mathbf{L}_j \mathbf{p}_j$ in the objective function (6.5) is exactly the normalized graph cut [134, 87], whose minimization leads to a two-way clustering with minimum intra-cluster connections [134]. In particular, this term can be further divided into two parts, i.e.,

$$\mathbf{p}_j^\top \mathbf{L} \mathbf{p}_j = \sum_{i,i'} \left[s_{i,i'} (p_{i,j} - p_{i',j})^2 / \sum_i s_{i,i'} \right] \tag{6.6}$$

Thus, by minimizing the first term, we enforce the consistency between the class label assignment and the similarity measurement. The second term $(\mathbf{p}_j - \hat{\mathbf{p}}_j)^\top \mathbf{\Lambda}_j (\mathbf{p}_j - \hat{\mathbf{p}}_j)$ is a term that penalize the disagreement between the estimated probabilities $\mathbf{p}_j$ with $\hat{\mathbf{p}}_j$, the class labels of the seed examples. Instead of formulating it as a hard constraint in the optimization problem, we use the penalty term to ensure the consistency between $\mathbf{p}_j$ and $\hat{\mathbf{p}}_j$. The advantage of such a treatment is that it leaves the room for correcting labeling mistakes in the seed examples.

It is easy to find our proposed model in the expression (6.5) can be viewed as a multi-class version of the Spectral Graph Transducer model reviewed in Section 6.2. As pointed out before, our proposed model can be explained from the view of label propagation: the enforcement of within page format consistency and agreement with seed examples on all the probabilities $p_{i,j}$ can be viewed as propagating the labeling information from the labeled seed text segments to the unlabeled ones along the structure of the weighted graph. Since we need to decide the probabilities of assigning each text segment to three classes, our approach indeed has three separate propagations, each for a different class. It is important to point out that the three propagation processes are strongly correlated. This is because if a text segment is assigned with a large probability for one class, the probability of assigning the text segment to the other classes has to be small. This can be further illustrated by the

Step 0	Divide the page into text segments and compute their pairwise format similarity (as described in Section 6.5.1)
Step 1	Initialize a labeled example set $\mathcal{L}$ by identifying a few seed examples (as described in Section 6.4)
Step 1.1	Solve the semi-supervised model in (6.5)
Step 1.2	Predict the class label of each text segment by the class with the largest probability.

Figure 6.5. The semi-supervised learning algorithm for extracting QA pairs from FAQ pages

dual problem of (6.5), i.e.,

$$\min_{\mathbf{u},\lambda} \sum_{j=1}^{3} (\lambda + \mathbf{u}_j + C_j \mathbf{\Lambda}_j \hat{\mathbf{p}}_j)^\top (\mathbf{L} + C_j \mathbf{\Lambda}_j)^{-1} (\lambda + \mathbf{u}_j + C_j \mathbf{\Lambda}_j \hat{\mathbf{p}}_j)$$
$$\text{s. t. } \mathbf{u}_j \geq 0,\ j = 1, 2, 3$$

where λ and $\mathbf{u}_j$ are the Lagrangian multipliers. Given the optimal solution for λ and $\mathbf{u}_j$, the solution for the primal problem can be written as:

$$\mathbf{p}_j = (\mathbf{L} + C_j \mathbf{\Lambda}_j)^{-1} (\lambda + \mathbf{u}_j + C_j \mathbf{\Lambda}_j \hat{\mathbf{p}}_j)$$

Clearly, the solutions for all $\mathbf{p}_j$'s are correlated through the Lagrangian multiplier λ. Finally, compared to the SGT approach, another advantage of the proposed algorithm is its probabilistic nature. As we will show in our experiments, the estimated probability does reveal the uncertainty in classifying text segments into the class of questions, answers, or noise text.

The optimization problem (6.5) is a quadratic programming problem, and therefore can be solved efficiently using the standard packages. Given the estimated probabilities in $\mathbf{P}$, we predict the class of each text segment by the one with the largest probability. Figure 6.5 summarizes the proposed algorithm for QA extraction.

As the final step, we need to form the QA pairs based on the class labels assigned to the text segments in each FAQ page. To this end, we first reduce the sequence

by removing all the text segments being labeled as "noise text"; then in the reduced sequence, we merge those consecutive text segments that have the same label; finally, we pair each merged text segment being labeled as answers with its most immediate proceeding text segment being labeled as questions, thus form a QA pair. For example, suppose we have a sequence of text segments (represented by their IDs) and their classes labels (represented by `Q, A, O`) as follows

```
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
O  O  Q  Q  O  A  A  O  Q  A  A  O  A  Q  A  A  O
```

Then three QA pairs can be extracted from the above sequence:

```
QA Pair 1: Question-#3,#4;   Answer-#6,#7
QA Pair 2: Question-#9;      Answer-#10,#11,#13
QA Pair 3: Question-#14;     Answer-#15,#16
```

6.6 Experiments and Discussions

6.6.1 Experiment Setup

Our testbed includes 10,912 text segments, which forms 1303 QA pairs [5]. All these 1303 QA pairs are all manually labeled, and will be used as the truth for evaluation. Since our proposed algorithm does not need any training data, we will carry out our proposed algorithm on all the FAQ pages, and compare the extracted QA pairs with the truth, i.e., the manual labeled QA pairs, for evaluation.

Both precision and recall are used as evaluation metrics. They are defined as

[5]These text segments come from 100 FAQ pages that are randomly selected from those FAQ pages we've identified from our web crawl data, as described in Section 6.3. The selection does not favor any particular domain or page content). The size of this dataset is comparable to those used in [99, 136, 84, 137].

follows

$$\text{Precision} = \frac{\text{Number of correctly extracted QA pairs}}{\text{Number of extracted QA pairs}}$$
$$\text{Recall} = \frac{\text{Number of correctly extracted QA pairs}}{\text{Number of the true QA pairs}}$$

Note that a QA pair is "correctly extracted" if and only if the corresponding question and answer share the *identical* sets of text segments with those that are manually identified as a QA pair.

Since the above evaluation metrics defined above view all the text of an QA pair as a whole, we refer to it as evaluation **by QA**. Evidently, these evaluation metrics are very "strict" in that as long as an extracted QA pair is not identical to the true one, it won't count as correct, no matter how small the extraction error may be. Here, we define another set of metrics based on word counting that are relatively "looser" compared to the evaluation by QA. Specifically, we can define

$$\text{Precision} = \frac{\text{Number of correctly extracted QA words}}{\text{Number of extracted QA words}}$$
$$\text{Recall} = \frac{\text{Number of correctly extracted QA words}}{\text{Number of words in the true QA pairs}}$$

Here, a word is correctly extracted as long as it is in the true QA pairs. Note that using this set of evaluation metrics, an extracted QA pair will have some chance to obtain "partial credit" when it is not identical to the true one. Since the evaluation is based on word counting, we call it as evaluation **by word**. Furthermore, we combine precision and recall together and compute the $F1$ score that is defined as harmonic mean of precision and recall. All the above precision/recall/$F1$ metrics for QA pair extraction can be extended to evaluate the performance of extracting FAQ questions or FAQ answers separately.

To obtain comprehensive understanding of the overall performance of the extraction, two different averaging methods are used in our evaluation. In the first averaging method, we compute the evaluation metric (i.e., precision, recall, or $F1$)

for each FAQ page, and average it across all FAQ pages. We refer to this averaging as *macro-averaging*. Alternatively, we can directly compute the evaluation metrics for all the individual QA pairs from all the FAQ pages. We refer to this method as *micro-averaging*.

We implemented three baseline methods. The first two are based on heuristics, and have been used in previous studies [99, 136, 84, 137]. Both baseline methods employ an optimal set of heuristics[6] to first identify all the questions. In the first baseline method, all the text between two consecutive questions are identified as the answers to the proceeding question[7]. We refer to this method as "**H-all**". In the second baseline method, up to 3 sentences following each FAQ question are extracted as the corresponding FAQ answer. We refer to this baseline method as "**H-3**". The third baseline is Support Vector Machine with precomputed kernels, which is a representative supervised learning method. We use the 1/4 of the FAQ pages as the training set, and the rest as the test set. The kernel is precomputed using the similarity measurement defined in Section 6.5.1. LibSVM [34] software package is used in our experiment. We will refer to this method as "**kSVM**". Finally, we will use "**SSL**" to refer to the proposed semi-supervised learning algorithm for extracting QA pairs from FAQ pages. The constants in the optimization problem (6.5) are set as $C_1 = 1, C_2 = C_3 = 0.5$ from empirical experience.

6.6.2 Experiments on Verification of Side Information

We first examine the *Within Page Consistency* principle, namely two text segments of the same page tend to follow the same HTML format pattern if they are in the same class. To verify this principle, we computes two quantities for each pair of text segments using the manually labeled FAQ page set: the similarity between the

[6]The optimal set of heuristics are identified empirically.

[7]For the last question in a FAQ page, up to n text segments are extracted for its answer, where n is the average number of text segments in all previous FAQ answers.

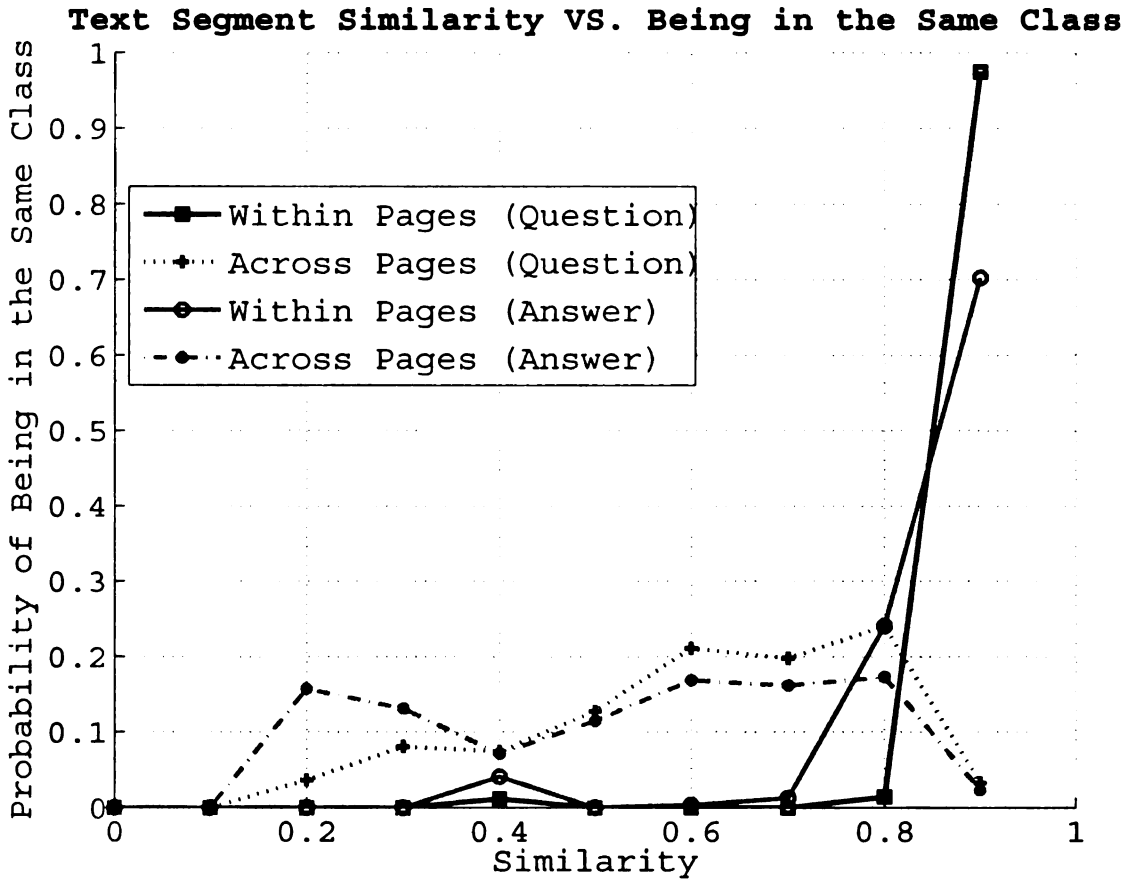


Figure 6.6. The probability distribution for two text segments to be in the same class versus the format similarity between text segments

two text segments in their HTML format, and whether or not the two segments are in the same class. Figure 6.6 summarizes the results by showing how the format similarity between two text segments affects the probability for them to be in the same class. The two solid lines show the results for the intra-class similarity of two text segments within the same pages, and the two dot lines show the results for the intra-class similarity of two text segments from different pages. It is clear that for the text segments of the same web pages, their intra-class similarity nicely indicate their relationship, namely the larger the similarity between two text segments, the more likely the two segments will be classified into the same class. In contrast, the intra-class similarity between text segments of different web pages appears to be uninformative to their class memberships. Both dot lines in Figure 6.6 are flat

across wide range of the similarity, and do not show the clear trend that the larger the similarity the higher the probability. We thus conclude that the *Within Page Consistency* principle provides a piece of valid side information for our experiment data.

6.6.3 Experiments on QA Extraction

Table 6.2 lists the precision/recall/$F1$ of the four different approaches for QA extraction. As shown in the table, the $F1$ scores of the proposed "SSL" method are always considerably better than the three baseline models, whether using the micro-averaging or the macro-averaging method. Especially, when using the strict evaluation metrics "by QA", the improvement made by the proposed algorithm is significant. Therefore, we can conclude that our proposed "SSL" method perform better than the baseline methods in extracting QA pairs from the FAQ pages.

Further analysis on the experiment results reveals the following findings

1. For all the four algorithms, the performance of FAQ question extraction is in general significantly better than that of FAQ answer extraction when using the metric of evaluation by QA. This is in accordance with the fact that answers are more challenging to extract due to its longer length and more diversified formats.
2. In general, for all the four extraction methods, the scores of evaluation by word are considerably higher than the scores of evaluation by QA. This observation implies that although the noise text mixed with the FAQ text is small in amount, it is *pervasive*. This is related to the fact that there are often repeated patterns in the FAQ pages, so an error in extracting one QA pair will be very likely to recur when extracting other QA pairs. When the extracted QA pairs are used in other applications as enumerated in Section 6.1, such kind of pervasive and recurring errors could be a very annoying factor that can seriously degrade the

		By QA					
		micro-averaging			macro-averaging		
		P	R	$F1$	P	R	$F1$
H-all	Pair	0.5055	0.4950	0.5002	0.5089	0.4919	0.4965
	Q	0.8706	0.6662	0.7548	0.8617	0.6564	0.7280
	A	0.5078	0.4973	0.5025	0.5105	0.4933	0.4980
H-3	Pair	0.4704	0.3599	0.4078	0.4601	0.3532	0.3910
	Q	0.8706	0.6662	0.7548	0.8617	0.6564	0.7280
	A	0.4794	0.3669	0.4157	0.4681	0.3578	0.3968
kSVM	Pair	0.1267	0.1696	0.1450	0.1001	0.1493	0.1189
	Q	0.3029	0.2304	0.2617	0.2109	0.1923	0.1956
	A	0.1323	0.1842	0.1540	0.1104	0.1569	0.1245
SSL	Pair	0.8163	0.7130	0.7612	0.7698	0.6838	0.7159
	Q	0.9411	0.8220	0.8775	0.9196	0.7891	0.8356
	A	0.8383	0.7322	0.7816	0.7901	0.7010	0.7344

		By Word					
		micro-averaging			macro-averaging		
		P	R	$F1$	P	R	$F1$
H-all	Pair	0.8728	0.6987	0.7761	0.8640	0.7518	0.8007
	Q	0.8896	0.7156	0.7932	0.9027	0.7196	0.7535
	A	0.8673	0.9051	0.8858	0.8559	0.9102	0.8680
H-3	Pair	0.9303	0.3543	0.5132	0.9137	0.4656	0.5790
	Q	0.8896	0.7156	0.7932	0.9027	0.7196	0.7535
	A	0.9441	0.3113	0.4682	0.9229	0.4365	0.5486
kSVM	Pair	0.4930	0.4265	0.4573	0.3922	0.3729	0.3654
	Q	0.5023	0.5313	0.5164	0.4238	0.4247	0.4156
	A	0.5123	0.4269	0.4657	0.4323	0.3920	0.4107
SSL	Pair	0.9806	0.8126	0.8887	0.9589	0.8077	0.8493
	Q	0.9800	0.8163	0.8907	0.9496	0.7981	0.8590
	A	0.9807	0.8727	0.9235	0.9506	0.8324	0.8827

Table 6.2. The performance of QA extraction by four different methods. The columns of "P", "R" and "F1" give the precision/recall/F1 scores.

application performance.

3. It is interesting to observe the performance gap between the "H-all" algorithm and the "SSL" when we use different evaluation metrics. Using the metrics of evaluation by word, the scores of the "H-all" algorithm are high, and sometimes close to those of the "SSL" algorithm. However when using the metrics of evaluation by QA, the performance gap between "H-all" and "SSL" is much larger. This phenomenon shows that instead of extracting a large number of QA pairs with small errors, our "SSL" method is able to yield many more *complete* and *noise-free* QA pairs, thus improving the quality of extracted QA text.

4. When comparing the "H-all" method with "H-3" method in terms of FAQ answer extraction [8], we observe that using the "by word" metrics, the former approach has a much better recall, but a worse precision. This is because the "H-all" algorithm includes all the text between two consecutive questions as the answer, and therefore achieves a high recall. In contrast, the "H-3" algorithm only includes the first up-to-three sentences following a question as the answer, and therefore achieves a high precision. This result shows the interesting trade-off between precision and recall that exists in most heuristics-based approaches.

5. The performance of "kSVM" method is extremely poor. This is not surprising, since the precomputed kernel encodes much information about the intra-class similarity from different pages, which is shown to be uninformative as discussed in Section 6.6.2. This experiment implies that the large variance existing in the presentation format from different FAQ pages is hard to be captured by a supervised learning algorithm, given a limited number of labeled examples.

As stated before, one important advantage of the proposed algorithm is that it

[8]Their performance in extracting FAQ questions are the same because the two methods only differ in their FAQ answer extraction strategies.

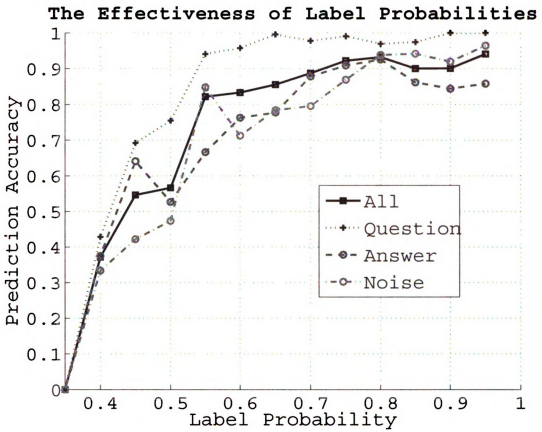


Figure 6.7. The correlation between prediction accuracy and classification probability estimated by the proposed algorithm for QA extraction

outputs not only the classification results but also the uncertainty in classification. To examine the quality of the class probabilities that are estimated by the proposed algorithm, for each text segment, we compute two quantities: the label probability output by the proposed algorithm, and whether or not the predicted class label is correct. Figure 6.7 summarizes these results by showing how the prediction accuracy of each class is affected by the estimated label probability. Overall speaking, the larger is the estimated probability, the higher is the prediction accuracy. We thus conclude that the estimated label probabilities do indicate the confidence of classifying text segments.

6.7 Conclusions

In this paper, we study the problem of automatically extracting QA pairs from web FAQs. We first identify the pattern diversity challenge, a key challenge in QA extraction from FAQ pages. We then present a semi-supervised learning algorithm that is effective in exploiting the *Within Page Consistency* principle, the key side information to the QA extraction task. The main advantage of the proposed algorithm is twofold: first, the proposed algorithm is able to boost the limited knowledge by generating the seed examples that are labeled by heuristics; second, the proposed algorithm is completely unsupervised. It predicts the class labels of text segments by propagating the class labels of seed examples to the other text segments of the same page. Empirical study shows that our proposed QA extraction method is able to yield more complete and noise-free extracted QA pairs, hence improving the extraction performance substantially over heuristics-based or supervised learning approaches in previous studies.

CHAPTER 7

Conclusions

The topic of this thesis work is semi-supervised learning with side information. In previous chapters, three generic learning tasks and two applications are discussed in details. The contributions, of each chapter respectively, can be summarized as follows

- In Chapter 2, we propose a constrained non-negative matrix factorization (CNMF) model for multi-label learning with class correlations, to meet the challenging situation of a large number of classes and a small size of training data.
- In Chapter 3, we propose a novel boosting framework, LinkBoost, to improve *any* supervised classification algorithm with link information.
- In Chapter 4, we propose a novel boosting framework, BoostCluster, to improve *any* clustering algorithm with pairwise constraints.
- In Chapter 5, we propose a novel statistical framework, Maximum Coherence Framework, for query translation disambiguation in cross-language information retrieval with a bilingual dictionary.
- In Chapter 6, we propose a semi-supervised learning model that utilizes human knowledge on web FAQs, to automatically extract question-answer pairs from them without human supervision.

The role of side information played in each task and application is listed as follows

- In CNMF model for multi-label learning, side information (in the form of class correlations) is encoded with class labels for computing pairwise example similarities, whose consistency with input pattern based similarities is further enforced.

- In LinkBoost framework for classification, side information (in the form of links) is combined with input pattern based example similarities, whose consistency with the pairwise relationship induced from class labels is further enforced.

- In BoostCluster framework for clustering, side information (in the form of pairwise constraints) is enforced to be consistent with input pattern based pairwise similarities computation.

- In Maximum Coherence model for CLIR, side information (in the form of dictionaries) is used to construct a graph, over which soft class memberships are enforced to be consistent with pairwise term similarities (or "coherence").

- In automatic question-answer extraction from web FAQs, side information (in the form of "within page consistency" knowledge) is used to correlate three class label propagations over the same graph, where each label propagation can be explained as consistency enforcement between class labels and input pattern based similarities.

From the above, it is easy to find that consistency enforcement is a common theme involved in the use of side information. On an abstract level, the rationale behind all the work presented in this thesis is the assumption that data examples that are close to each other, when judging either from their input patterns or side information, should be predicted similarly. This is a natural extension of the data consistency assumption underlying most graph-based learning approaches (as stated in Section 1.1.2).

To effectively use the side information against its usual nature of sparseness, incompleteness and noise, when incorporating side information into semi-supervised models, we deliberately avoid formulating them as hard constraints. Instead, we favor the use of side information in soft penalty. In this way, violation with side information is allowed, but with a loss; minimizing the collective violation loss results in tolerance with the noise in the side information. Also, all our proposed models inherit the spirit of graph-based learning that a connected graph is constructed over all the (labeled and unlabeled) data based on their input patterns. Such a graph serves as a good supplement to understand the underlying structure of data, wherever side information is missing. This treatment reduces the risk brought by the incompleteness and sparseness of side information. Consequently robustness is presented in all the proposed semi-supervised learning models with side information, as suggested by those experiments shown in previous chapters.

In all the semi-supervised models proposed in this thesis work, the theme of consistency enforcement is always achieved through optimizations. A comparison among all the objective functions in Chapter 2 through Chapter 6 will reveal a certain degree of resemblance. In particular, the consistency is always formalized in a pairwise manner, i.e., the overall consistency is decomposed into consistency measurements defined on each pair of data examples, including labeled and unlabeled ones. Such a formalization embodies the use of unlabeled data, and also fosters the propagation of supervised information from labeled data to unlabeled data. Another advantage of formalizing optimization objectives in a sum-of-pairwise manner is, by carefully choosing the consistency measurement defined in a pair of data examples, the resulting optimization is often convex, thus tractable and with global optimum.

The above analysis summarizes a few nice properties in the proposed semi-supervised models. In conclusion, the work presented in this thesis suggests a viable approach towards semi-supervised learning with side information: enforcing consis-

tency among data input patterns, supervised information (if any), side information, and predictions. We believe that this approach is applicable to a wide variety of learning tasks and application areas.

APPENDIX A

Related Proofs and Lists

A.1 Proof of the eigenvector problem in Section 4.3.3

We show that every non-zero eigenvector $\mathbf{v}_i$ can be written as a linear combination of $\tilde{\mathbf{x}}_i, i = 1, 2, \ldots, m$, i.e., the examples involved in the pairwise constraints. Let $\mathbf{v}$ and $\lambda \neq 0$ be an eigenvector and eigenvalue of matrix $\mathbf{XTX}^\top$. We therefore have $\mathbf{XTX}^\top\mathbf{v} = \lambda\mathbf{v}$. We further decompose $\mathbf{v}$ into two parts: $\mathbf{v} = \mathbf{v}_{\|} + \mathbf{v}_{\perp}$, where $\mathbf{v}_{\|}$ represents the projection of $\mathbf{v}$ in the subspace spanned by $\{\tilde{\mathbf{x}}_i\}_{i=1}^s$, and $\mathbf{x}_{\perp}$ represents the projection perpendicular to $\{\tilde{\mathbf{x}}_i\}_{i=1}^s$. To show $\mathbf{v}$ can be written as a linear combination of $\{\tilde{\mathbf{x}}_i\}_{i=1}^s$, we need to show $\mathbf{v}_{\perp} = \mathbf{0}$. To this end, we first utilize the expression in (4.20) to calculate $\mathbf{v}_{\perp}^\top\mathbf{XTX}^\top$, i.e.,

$$\mathbf{v}_{\perp}^\top\mathbf{XTX}^\top = \sum_{i,j=1}^{m} \tilde{T}_{i,j}\mathbf{v}_{\perp}^\top\tilde{\mathbf{x}}_i\tilde{\mathbf{x}}_j^\top = \mathbf{0}^\top$$

We then multiply the eigen equation $\mathbf{XTX}^\top\mathbf{v} = \lambda\mathbf{v}$ by $\mathbf{v}_{\perp}^\top$, which leads to the following equation

$$\mathbf{v}_{\perp}^\top\mathbf{XTX}^\top\mathbf{v} = 0 = \lambda\mathbf{v}_{\perp}^\top\mathbf{v} = \lambda\|\mathbf{v}_{\perp}\|_2^2$$

Since $\lambda \neq 0$, we have $\mathbf{v}_{\perp} = \mathbf{0}$ and $\mathbf{v} = \mathbf{v}_{\|}$.

A.2 Proof of the generalized eigenvector problem in Section 4.3.3

We will show that for the ith eigenvector $\mathbf{v}_i = \tilde{\mathbf{X}}\mathbf{w}_i$ of $\tilde{\mathbf{X}}\tilde{\mathbf{T}}\tilde{\mathbf{X}}^\top$, $\mathbf{w}_i$ corresponds to the ith eigenvector of the generalized eigenvector problem in (4.22). First, we realize that the orthogonality condition $\mathbf{v}_i^\top \mathbf{v}_j = \delta(i,j)$ becomes $\mathbf{w}_i^\top \tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top \mathbf{w}_j = \delta_{i,j}$. We can write the above condition for all $\mathbf{w}_i, i = 1, 2, \ldots, s$ in the matrix form, i.e., $\mathbf{W}^\top \tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top \mathbf{W} = \mathbf{I}_s$. Second, the eigenvectors $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_s)$ are the optimal solution to the following optimization problem, i.e.,

$$\begin{aligned} \underset{\mathbf{V}\in\mathbb{R}^{d\times s}}{\arg\max} \quad & \mathrm{tr}(\mathbf{V}^\top \mathbf{X}\mathbf{T}\mathbf{X}^\top \mathbf{V}) \\ \text{s. t.} \quad & \mathbf{V}^\top \mathbf{V} = \mathbf{I}_s \end{aligned}$$

Replacing $\mathbf{V}$ in the above optimization problem with $\mathbf{V} = \tilde{\mathbf{X}}\mathbf{W}$, we have

$$\begin{aligned} \max_{\mathbf{W}\in\mathbb{R}^{m\times s}} \quad & \mathrm{tr}(\mathbf{W}^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}\tilde{\mathbf{T}}\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}\mathbf{W}) \\ \text{s. t.} \quad & \mathbf{W}^\top \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}\mathbf{W} = \mathbf{I}_s \end{aligned}$$

It is well known that the optimal solution $\mathbf{W}$ to the above problem consists of the first s eigenvectors of the generalized eigenvector problem in (4.22).

A.3 Features for FAQ page classification

The 19 features we used in FAQ page classification are as follows

1. Occurrence of keywords `faq` or `faqs` in the URL host section [1]

2. Number of keywords `faq` or `faqs` in the URL query section

[1] A URL can be divided into six sections. For example in `http://www.amazon.com:83/search.html?q=travel#marker`, the *protocol* section is `http`, the *host* section is `www.amazon.com`, the *port* section is 83, the *path* section is `/search.html`, the *query* section is `q=travel`, and the *fragment* section is `marker`.

3. Number of keywords `faq` or `faqs` in the URL fragment section

4. Normalized position of keywords `faq` or `faqs` in the URL path section (for example, in URL path section `/education/faq/coins/denominations.shtml`, the keyword `faq` appears in the 3rd segment when counting from right to left. Since altogether there are 4 segments, the normalized position of the `faq` is 3/4)

5. Number of question marks in the page body text

6. Number of question marks in the page title

7. Number of question marks in the anchor text

8. Logarithm of total number of words in the whole page text

9. Number of anchors with question marks

10. Percentage of anchors-with-question-marks in all the anchors

11. Ratio of question marks to words in the page body text

12. Ratio of question marks to words in the page title

13. Ratio of question marks to words in all the anchor text

14. Average number of words between consecutive question marks

15. Number of text segments in the page body text

16. Number of text segments (see Section 6.5.1 for definition) with question marks in the page body text

17. Percentage of text-segments-with-question-marks in all the text segments

18. Percentage of links in all the text-segments-with-question-marks

19. Number of well separated text-segments-with-question-marks (two consecutive text segments are *well separated* if there are at least k words between them. In our practice, we set $k = 3$.)

BIBLIOGRAPHY

[1] Mirna Adriani. Dictionary-based clir for the clef multilingual track. In *Cross-Language Information Retrieval and Evaluation, Workshop of Cross-Language Evaluation Forum, CLEF 2000*, Lisbon, September 2000.

[2] Mirna Adriani. Using statistical term similarity for sense disambiguation in cross-language information retrieval. *Inf. Retr.*, 2(1):71–82, 2000.

[3] Eugene Agichtein, Steve Lawrence, and Luis Gravano. Learning to find answers to questions on the web. *ACM Transactions on Internet Technology*, 4(2):129–162, 2004.

[4] Yasemin Altun, Thomas Hofmann, and Alexander J. Smola. Gaussian process classification for segmenting and annotating sequences. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 4, New York, NY, USA, 2004. ACM Press.

[5] M. R. Anderberg. *Cluster Analysis for Applications.* Academic Press, Inc., New Youk, NY, 1973.

[6] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*, pages 561–568, 2002.

[7] Ralitsa Angelova and Gerhard Weikum. Graph-based text classification: learn from your neighbors. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 485–492, New York, NY, USA, 2006. ACM.

[8] Lisa Ballesteros and W. Bruce Croft. Phrasal translation and query expansion techniques for cross-language information retrieval. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 84–91. ACM Press, 1997.

[9] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning distance functions using equivalence relations. In *ICML '03: Proceedings of the Twentieth International Conference*, pages 11–18, August 21-24 2003.

[10] Kobus Barnard, Pinar Duygulu, David A. Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[11] Kobus Barnard and David A. Forsyth. Learning the semantics of words and pictures. In *ICCV '01: Proceedings. Eighth IEEE International Conference on Computer Vision*, pages 408–415, 2001.

[12] Sugato Basu. *Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments.* PhD thesis, The University of Texas at Austin, 2005.

[13] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Semi-supervised clustering by seeding. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 27–34, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[14] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Active semi-supervision for pairwise constrained clustering. In *SDM '04: Proc. of the Fourth SIAM International Conference on Data Mining*, 2004.

[15] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 59–68, New York, NY, USA, 2004. ACM Press.

[16] R. Bekkerman and M. Sahami. Semi-supervised clustering using combinatorial MRFs. In *Proc. of ICML-06 Workshop on Learning in Structured Output Spaces*, 2006.

[17] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[18] Mikhail Belkin and Partha Niyogi. Using manifold stucture for partially labeled classification. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 15*, Cambridge, MA, 2002. MIT Press.

[19] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. In Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press, 2006.

[20] K. Bennett, P. Bradley, and A. Demiriz. Constrained k-means clustering. Technical Report 2000-65, Microsoft Research, May 2000.

[21] Kristin P. Bennett, Ayhan Demiriz, and Richard Maclin. Exploiting unlabeled data in ensemble methods. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–296, New York, NY, USA, 2002. ACM.

[22] Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the lexical chasm: statistical approaches to answer-finding. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 192–199, New York, NY, USA, 2000. ACM Press.

[23] Adam Berger and Vibhu O. Mittal. Query-relevant summarization using faqs. In *ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 294–301, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

[24] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 11, New York, NY, USA, 2004. ACM Press.

[25] Avrim Blum and Shuchi Chawla. Learning from labeled and unlabeled data using graph mincuts. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 19–26, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[26] Avrim Blum, John Lafferty, Mugizi Robert Rwebangira, and Rajashekar Reddy. Semi-supervised learning using randomized mincuts. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 13, New York, NY, USA, 2004. ACM Press.

[27] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, New York, NY, USA, 1998. ACM.

[28] Matthew R. Boutella, Jiebo Luob, Xipeng Shen, and Christopher M. Browna. Learning multi-label scene classification. *Pattern Recognition*, 37:1757–1771, 2004.

[29] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[30] Robin D. Burke, Kristian J. Hammond, Vladimir A. Kulyukin, Steven L. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently asked

question files: Experiences with the faq finder system. Technical report, University of Chicago, Chicago, IL, USA, 1997.

[31] Lijuan Cai and Thomas Hofmann. Hierarchical document categorization with support vector machines. In *Proc. ACM Conf Information and Knowledge Management*, 2004.

[32] Pavel Calado, Marco Cristo, Edleno Moura, Nivio Ziviani, Berthier Ribeiro-Neto, and Marcos Andre; Goncalves. Combining link-based and content-based methods for web document classification. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 394–401, New York, NY, USA, 2003. ACM Press.

[33] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 307–318, New York, NY, USA, 1998. ACM.

[34] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[35] Zheng Chen, Liu Wenyin, Feng Zhang, and Mingjing Li. Web mining for web image retrieval. *J. Am. Soc. Inf. Sci. Technol.*, 52(10):831–839, 2001.

[36] Hai Leong Chieu and Hwee Tou Ng. A maximum entropy approach to information extraction from semi-structured and free text. In *Eighteenth national conference on Artificial intelligence*, pages 786–791, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence.

[37] Fan R. K. Chung. Eigenvalues of graphs. In *Proceedings of the International Congress of Mathematicians*, pages 1333–1342, Zurich, 1994. Birkh"auser Verlag, Berlin.

[38] Fan R. K. Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics, ISSN: 0160-7642. American Mathematical Society, 1997.

[39] William W. Cohen. Improving a page classifier with anchor extraction and link analysis. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1481–1488. MIT Press, Cambridge, MA, 2002.

[40] David Cohn, Deepak Verma, and Karl Pfleger. Recursive attribute factoring. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 297–304. MIT Press, Cambridge, MA, 2007.

[41] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-agressive algorithms. *Machine Learning*, 7:551–585, 2006.

[42] Koby Crammer and Yoram Singer. A new family of online algorithms for category ranking. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2002.

[43] Florence d'Alché Buc, Yves Grandvalet, and Christophe Ambroise. Semi-supervised marginboost. In *Advances in Neural Information Processing Systems 14*, pages 553–560, 2001.

[44] I. Davidson and S.S. Ravi. Clustering under constraints: Feasibility results and the k-means algorithm. In *SIAM Data Mining Conference*, 2005.

[45] I. Davidson and S.S. Ravi. Hierarchical clustering with constraints: Theory and practice. In *Proc. of the 9th European Principles and Practice of KDD (PKDD)*, 2005.

[46] Mark W. Davis. New experiments in cross-language text retrieval at NMSU's computing research lab. In D. K. Harman, editor, *The Fifth Text REtrieval Conference (TREC-5)*. NIST, 1996.

[47] Brian D. Davison. Topical locality in the web. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 272–279, New York, NY, USA, 2000. ACM Press.

[48] Brian D. Davison. Toward a unification of text and link analysis. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 367–368, New York, NY, USA, 2003. ACM Press.

[49] 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 *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 2001)*, pages 107–114. IEEE Computer Society, 2001.

[50] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[51] Abdessamad Echihabi and Daniel Marcu. A noisy-channel approach to question answering. In *ACL '03: Proceedings of the 41st Annual Meeting of the Association for Computation al Linguistics*, Sapporo, Japan, 2003.

[52] Andre Elisseeff and JasonWeston. A kernel method for multi-labelled classification. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 681–687, Cambridge, MA, 2002. MIT Press.

[53] Marcello Federico and Nicola Bertoldi. Statistical cross-language information retrieval using N-best query translations. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 167–174. ACM Press, 2002.

[54] Christiane Fellbaum, editor. *Wordnet : an electronic lexical database.* MIT Press, 1998.

[55] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. pages 23–37, 1995.

[56] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences*, 55(1):119–139, 1997.

[57] Jianfeng Gao, Jian-Yun Nie, Endong Xun, Jian Zhang, Ming Zhou, and Changning Huang. Improving query translation for cross-language information retrieval using statistical models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–104. ACM Press, 2001.

[58] Jianfeng Gao, Ming Zhou, Jian-Yun Nie, Hongzhao He, and Weijun Chen. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 183–190. ACM Press, 2002.

[59] Sheng Gao, Wen Wu, Chin-Hui Lee, and Tat-Seng Chua. A MFoM learning approach to robust multiclass multi-label text categorization. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, 2004.

[60] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 195–200, New York, NY, USA, 2005. ACM Press.

[61] Rayid Ghani, Seán Slattery, and Yiming Yang. Hypertext categorization using hyperlink patterns and meta data. In *ICML '01: Proceedings of the Eighteenth*

International Conference on Machine Learning, pages 178–185, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[62] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, Inc., San Diego, USA, 1981.

[63] Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, and Gary W. Flake. Using web structure for classifying and describing web pages. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 562–569, New York, NY, USA, 2002. ACM.

[64] Jacob Goldberger, Sam T. Roweis, Geoffrey E. Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, 2004.

[65] Gene H. Golub and Charles F. Van Loan. *Matrix Computation.* John Hopkins Press, 1989.

[66] Ming Gu, Hongyuan Zha, Chris Ding, Xiaofeng He, and Horst Simon. Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering. Technical Report CSE-01-007, Department of Computer Science and Engineering, Pennsylvania State University, 2001.

[67] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 265–272, New York, NY, USA, 2005. ACM Press.

[68] L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE. Trans. on Computed Aided Desgin*, 11:1074–1085, 1992.

[69] Sanda Harabagiu and Finley Lacatusu. Strategies for advanced question answering. In Sanda Harabagiu and Finley Lacatusu, editors, *HLT-NAACL 2004: Workshop on Pragmatics of Question Answering*, pages 1–9, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

[70] Taher H. Haveliwala. Topic-sensitive pagerank. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 517–526, New York, NY, USA, 2002. ACM Press.

[71] Tomer Hertz, Aharon Bar-Hillel, and Daphna Weinshall. Boosting margin based distance functions for clustering. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 50, New York, NY, USA, 2004. ACM Press.

[72] Tomer Hertz, Aharon Bar Hillel, and Daphna Weinshall. Learning a kernel function for classification with small training samples. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 401 -408, New York, NY, USA, 2006. ACM Press.

[73] Tomer Hertz, Noam Shental, Aharon Bar-Hillel, and Daphna Weinshall. Enhancing image and video retrieval: Learning via equivalence constraints. In *CVPR '03: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 02, page 668, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

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

[75] Stefan Holland, Martin Ester, and Werner Kießling. Preference mining: A novel approach on mining user preferences for personalized applications. In *PKDD*, pages 204–216, 2003.

[76] David A. Hull and Gregory Grefenstette. Querying across languages: a dictionary-based approach to multilingual information retrieval. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–57. ACM Press, 1996.

[77] ImageCLEF. The CLEF Cross Language Image Retrieval Track (ImageCLEF), http://ir.shef.ac.uk/imageclef/, 2003.

[78] Abraham Ittycheriah, Martin Franz, and Salim Roukos. IBM's statistical question answering system. In *TREC '01: Proceedings of the 10th Text REtrieval Conference*, Gaithersburg, MD, 2001.

[79] Tommi S. Jaakkola. *Variational methods for inference and estimation in graphical models.* PhD thesis, MIT, 1997. Supervisor-Michael I. Jordan.

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

[81] Myung-Gil Jang, Sung Hyon Myaeng, and Se Young Park. Using mutual information to resolve query translation ambiguities and query term weighting. In *ACL '99: Proceedings of the 37th annual meeting of the association for computational linguistics*, 1999.

[82] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding semantically similar questions based on their answers. In *SIGIR '05: Proceedings of the 28th*

annual international ACM SIGIR conference on Research and development in information retrieval, pages 617–618, New York, NY, USA, 2005. ACM Press.

[83] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding similar questions in large question and answer archives. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 84–90, New York, NY, USA, 2005. ACM Press.

[84] Valentin Jijkoun and Maarten de Rijke. Retrieving answers from frequently asked questions pages on the web. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83, New York, NY, USA, 2005. ACM Press.

[85] Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 897–904. MIT Press, Cambridge, MA, 2003.

[86] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *Proc European Conference on Machine Learning*, 1998.

[87] Thorsten Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.

[88] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, 1999.

[89] Sepandar D. Kamvar, Dan Klein, and Christopher D. Manning. Spectral Learning. In *IJCAI '03: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, August 9-15 2003.

[90] Hideto Kazawa, Tomonori Izumitani, Hirotoshi Taira, and Eisaku Maeda. Maximal margin labeling for multi-topic text categorization. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 649–656. MIT Press, Cambridge, MA, 2005.

[91] D. Klein, S. D. Kamvar, and C. D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML '02: Proceedings of 19th Intl. Conf. on Machine Learning*, 2002.

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

[93] W. Kraaij, R. Pohlmann, and D. Hiemstra. Twenty-one at TREC-8: using language technology for information retrieval. In Ellen M. Voorhees and Donna K. Harman, editors, *The Eighth Text REtrieval Conference (TREC-8)*, volume 8, pages 285–300. National Institute of Standards and Technology, NIST, 2000. NIST Special Publication 500-246.

[94] Wessel Kraaij, Jian-Yun Nie, and Michel Simard. Embedding web-based statistical translation models in cross-language information retrieval. *Comput. Linguist.*, 29(3):381–419, 2003.

[95] Wessel Kraaij and Renée Pohlmann. Different approaches to cross language information retrieval. In W. Daelemans, K. Sima'an, J. Veenstra, and J. Zavrel, editors, *Computational Linguistics in the Netherlands 2000*, number 37 in Language and Computers: Studies in Practical Linguistics, pages 97–111, Amsterdam, 2001. Rodopi.

[96] Brian Kulis, Sugato Basu, Inderjit Dhillon, and Raymond Mooney. Semi-supervised graph clustering: a kernel approach. In *ICML '05: Proc. of the 22nd international conference on Machine learning*, pages 457–464, New York, NY, USA, 2005. ACM Press.

[97] Vladimir A. Kulyukin, Kristian A. Hammond, and Robin D. Burke. Automated processing of structured online documents. Technical report, University of Chicago, Chicago, IL, USA, 1998.

[98] James T. Kwok and Ivor W. Tsang. Learning with idealized kernels. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pages 400–407, Washington, D.C., USA, August 2003.

[99] Yu-Sheng Lai, Kuao-Ann Fung, and Chung-Hsien Wu. Faq mining via list detection. In *COLING-02: proceeding of the 2002 conference on multilingual summarization and question answering*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[100] Victor Lavrenko, Martin Choquette, and W. Bruce Croft. Cross-lingual relevance models. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 175–182. ACM Press, 2002.

[101] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM Press, 2001.

[102] Martin H. C. Law, Alexander P. Topchy, and Anil K. Jain. Model-based clustering with probabilistic constraints. In *SIAM International Conference on Data Mining (SDM)*, 2005.

[103] Neil D. Lawrence and Michael I. Jordan. Semi-supervised learning via gaussian processes. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 753–760. MIT Press, Cambridge, MA, 2005.

[104] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.

[105] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562, 2000.

[106] Shuang Liu, Fang Liu, Clement Yu, and Weiyi Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 266–272, New York, NY, USA, 2004. ACM Press.

[107] Qing Lu and Lise Getoor. Link-based classification. In *ICML '03: Proceedings of the twentieth international conference on Machine learning*, pages 496–503, 2003.

[108] Zhengdong Lu and Todd Leen. Semi-supervised learning with penalized probabilistic clustering. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 849–856, Cambridge, MA, 2005. MIT Press.

[109] Akira Maeda, Fatiha Sadat, Masatoshi Yoshikawa, and Shunsuke Uemura. Query term disambiguation for web cross-language information retrieval using a search engine. In *Proceedings of the 5 th International Workshop on Information Retrieval with Asian Languages (IRAL '00)*, pages 25–32. ACM Press, 2000.

[110] Andrew McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI'99 Workshop on Text Learning*, 1999.

[111] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[112] Joel C. Miller, Gregory Rae, Fred Schaefer, Lesley A. Ward, Thomas LoFaro, and Ayman Farahat. Modifications of kleinberg's hits algorithm using matrix exponentiation and web log records. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 444–445, New York, NY, USA, 2001. ACM Press.

[113] Tom Mitchell. *Machine Learning.* McGraw Hill, 1997.

[114] Christof Monz and Bonnie J. Dorr. Iterative translation disambiguation for cross-language information retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 520–527, 2005.

[115] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856, 2001.

[116] Jian-Yun Nie and Michel Simard. Using statistical translation models for bilingual ir. In *Cross-Language Information Retrieval and Evaluation, Workshop of Cross-Language Evaluation Forum, CLEF 20'01*, pages 137–150. Springer-Verlag, 2002.

[117] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[118] Hyo-Jung Oh, Sung Hyon Myaeng, and Mann-Ho Lee. A practical hypertext catergorization method using links and incrementally available class information. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 264–271, New York, NY, USA, 2000. ACM.

[119] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[120] Xiaoguang Qi and Brian D. Davison. Knowing a web page by the company it keeps. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 228–237, New York, NY, USA, 2006. ACM.

[121] Dragomir R. Radev, Hong Qi, Zhiping Zheng, Sasha Blair-Goldensohn, Zhu Zhang, Weigo Fan, and John Prager. Mining the web for answers to natural language questions. In *CIKM '01: Proceedings of the 10th ACM international conference on Information and knowledge management*, Atlanta, GA, 2001.

[122] Ganesh Ramakrishnan, Soumen Chakrabarti, Deepa Paranjpe, and Pushpak Bhattacharya. Is question answering an acquired skill? In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 111–120, New York, NY, USA, 2004. ACM Press.

[123] Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor. On maximum margin hierarchical multi-label classification. In *NIPS Workshop on Learning With Structured Outputs*, 2004.

[124] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. In *Science*, 2000.

[125] Walter Rudin. *Real & Complex Analysis.* McGraw-Hill, Inc., New York, NY, 3rd edition, 1987.

[126] Gerard Salton, editor. *The SMART retrieval system.* Prentice-Hall, 1971.

[127] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based systemfor text categorization. *Machine Learning*, 39(2-3), 2000.

[128] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems 16*, 2003.

[129] M. Seeger. Learning with labeled and unlabeled data. Technical report, University of Edinburgh., 2001.

[130] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. A comparison of implicit and explicit links for web page classification. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 643–650, New York, NY, USA, 2006. ACM.

[131] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, New York, NY, USA, 2005. ACM Press.

[132] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit user modeling for personalized search. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831, New York, NY, USA, 2005. ACM Press.

[133] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with em using side-information. In *Proc. of workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, 2003.

[134] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. on PAMI*, 22(8):888–905, August 2000.

[135] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 21–29, New York, NY, USA, 1996. ACM Press.

[136] Radu Soricut and Eric Brill. Automatic question answering: Beyond the factoid. In *HLT-NAACL 2004, Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 57–64, 2004.

[137] Radu Soricut and Eric Brill. Automatic question answering using the web: Beyond the factoid. *Information Retrieval - Special Issue on Web Information Retrieval*, 9(2):191–206, 2006.

[138] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *ICML '03: Proceedings of the twentieth international conference on Machine learning*, pages 720–727, 2003.

[139] Smitha Sriram, Xuehua Shen, and Chengxiang Zhai. A session-based search engine. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 492–493, New York, NY, USA, 2004. ACM Press.

[140] Jian-Tao Sun, Hua-Jun Zeng, Huan Liu, Yuchang Lu, and Zheng Chen. Cubesvd : a novel approach to personalized web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 382–390, New York, NY, USA, 2005. ACM Press.

[141] Ben Taskar, Vassil Chatalbashev, and Daphne Koller. Learning associative markov networks. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 102, New York, NY, USA, 2004. ACM Press.

[142] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000.

[143] The Mathworks. Matlab, http://www.mathworks.com/.

[144] John A. Tomlin. A new paradigm for ranking pages on the world wide web. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 350–355, New York, NY, USA, 2003. ACM Press.

[145] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 104, New York, NY, USA, 2004. ACM Press.

[146] Naonori Ueda and Kazumi Saito. Parametric mixture models for multi-labeled text. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 15*, pages 649–656. MIT Press, Cambridge, MA, 2003.

[147] Ellen M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 171–180, New York, NY, USA, 1993. ACM Press.

[148] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[149] Jidong Wang, Huajun Zeng, Zheng Chen, Hongjun Lu, Li Tao, and Wei-Ying Ma. Recom : reinforcement clustering of multi-type interrelated data objects. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 274–281, New York, NY, USA, 2003. ACM.

[150] Kilian Weinberger, John Blitzer, and Lawrence Saul. Distance metric learning for large margin nearest neighbor classification. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1473–1480, Cambridge, MA, 2006. MIT Press.

[151] Eric W. Weisstein. "graph." from mathworld–a wolfram web resource. http://mathworld.wolfram.com/graph.html.

[152] C. Williams. Computation with infinite neural networks. *Neural Computation*, 10(5), 1998.

[153] Chung-Hsien Wu, Jui-Feng Yeh, and Ming-Jun Chen. Domain-specific faq retrieval using independent aspects. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(1):1–17, 2005.

[154] Chung-Hsien Wu, Jui-Feng Yeh, and Yu-Sheng Lai. Semantic segment extraction and matching for internet faq retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 18(7):930–940, 2006.

[155] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512, Cambridge, MA, 2003. MIT Press.

[156] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, Zurich, Switzerland, 1996.

[157] Jinxi Xu and Ralph Weischedel. TREC-9 cross-lingual retrieval at BBN. In *The Ninth Text REtrieval Conference (TREC-9)*, 2001.

[158] Gui-Rong Xue, Dou Shen, Qiang Yang, Hua-Jun Zeng, Zheng Chen, Yong Yu, WenSi Xi, and Wei-Ying Ma. Irc : An iterative reinforcement categorization algorithm for interrelated web objects. In *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 273–280, Washington, DC, USA, 2004. IEEE Computer Society.

[159] Liu Yang, Rong Jin, Rahul Sukthankar, and Yi Liu. An efficient algorithm for local distance metric learning. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, 2006.

[160] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2), 1999.

[161] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49, New York, NY, USA, 1999. ACM Press.

[162] Yiming Yang, Seán Slattery, and Rayid Ghani. A study of approaches to hypertext categorization. *J. Intell. Inf. Syst.*, 18(2-3):219–241, 2002.

[163] Jieping Ye. Generalized low rank approximations of matrices. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 112, New York, NY, USA, 2004. ACM Press.

[164] Kai Yu, Shipeng Yu, and Volker Tresp. Multi-label informed latent semantic indexing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2005.

[165] Z. Zhang, J.T. Kwok, and D.Y. Yeung. Parametric distance metric learning with label information. In *Proc. of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03)*, pages 1450–1452, 2003.

[166] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[167] Dengyong Zhou, Jiayuan Huang, and Bernhard Scholkopf. Learning from labeled and unlabeled data on a directed graph. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 1036–1043, New York, NY, USA, 2005. ACM Press.

[168] Dengyong Zhou, Bernhard Schölkopf, and Thomas Hofmann. Semi-supervised learning on directed graphs. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1633–1640. MIT Press, Cambridge, MA, 2005.

[169] Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on data manifolds. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[170] Xiang Sean Zhou and Thomas S. Huang. Unifying keywords and visual contents in image retrieval. *IEEE MultiMedia*, 9(2):23–33, 2002.

[171] Shenghuo Zhu, Xiang Ji, Wei Xu, and Yihong Gong. Multi-labelled classification using maximum entropy method. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 274–281, New York, NY, USA, 2005. ACM Press.

[172] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005. http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.

[173] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML '03: Proceedings of the twentieth international conference on Machine learning*, pages 912–919, 2003.

[174] Xiaojin Zhu and John Lafferty. Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 1052–1059, New York, NY, USA, 2005. ACM Press.