The Dissertation Committee for Sugato Basu

certifies that this is the approved version of the following doctoral dissertation:

Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments

Committee:

Raymond J. Mooney, Supervisor

Inderjit S. Dhillon

Joydeep Ghosh

C. Gregory Plaxton

Mehran Sahami

Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments

by

Sugato Basu, B. Tech (Hons); M.S.

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2005

In loving memory of my grandparents

Acknowledgments

First and foremost, I would like to thank my advisor. It was a real pleasure working with Ray. He was patient and gave me a lot of freedom when I was stumbling around looking for a research problem in the early years, he kept my spirits high with his motivation whenever anything went wrong, he spent a remarkable amount of time mentoring me on a personal level, he taught me not only how to be a successful PhD student but how to do good science – in short, he was an incredible mentor. Wherever I go from here, I would definitely miss my weekly chats with Ray, from which I always came out recharged.

I would like to thank my thesis committee members, Joydeep, Inderjit, Greg and Mehran, for giving me extremely valuable feedback on my work. I learnt a lot from them through our interesting research discussions. Thanks are specially due to Mehran for being a wonderful mentor and friend, and for making my time at Google so enjoyable during my two summers there.

I was truly lucky to have a remarkable set of collaborators at UT Austin, especially Arindam and Misha – I could not have found more motivated, brilliant, helpful and friendly people to work with. I have learnt so much in the long hours spent discussing research with all my fellow UT graduate students, and thoroughly enjoyed dissecting papers and throwing around ideas at ULG. I would like to thank the Machine Learning group members for making our group such a nice and friendly place to work in.

I would like to thank the wonderful staff in our department, especially Gloria, Katherine, Laurie and Stacy – I would have missed so many registration deadlines and conference travels if not for your help and support.

I would like to thank all my friends at UTBC, especially Sandipda – I will truly treasure some of the friendships I formed here in Austin.

Finally, I would like to thank my family for their contribution in my PhD. The love, affection and inspiration of Baba, Ma, Putu has shaped my life; I cannot be grateful enough for their untiring support and unconditional belief in me. I am lucky to have Kaku, Kakima, Bhai in my life – their love has made me stronger, their support has helped me through so many hard times. The love I have got and still get from all my cousins, uncles and aunts makes me feel lucky to be part of such a wonderful family.

Last but not the least, I could not have done any of this without my best friend, my philosopher and guide – my wife Shalini. She gave me hope whenever I was down, direction when I got lost, strength when I felt weak, and unquestioning love all along. She is the best thing that has ever happened to me – the smartest move in my life was snagging her when she was sixteen.

SUGATO BASU

The University of Texas at Austin

August 2005

Semi-supervised Clustering: Probabilistic Models, Algorithms and Experiments

Publication No. ______________________

Sugato Basu, Ph.D.
The University of Texas at Austin, 2005

Supervisor: Raymond J. Mooney

Clustering is one of the most common data mining tasks, used frequently for data categorization and analysis in both industry and academia. The focus of our research is on semi-supervised clustering, where we study how prior knowledge, gathered either from automated information sources or human supervision, can be incorporated into clustering algorithms. In this thesis, we present probabilistic models for semi-supervised clustering, develop algorithms based on these models and empirically validate their performances by extensive experiments on data sets from different domains, e.g., text analysis, hand-written character recognition, and bioinformatics.

In many domains where clustering is applied, some prior knowledge is available either in the form of labeled data (specifying the category to which an instance belongs) or pairwise constraints on some of the instances (specifying whether two instances should be in same or different clusters). In this thesis, we first analyze effective methods of incorporating labeled supervision into prototype-based clustering algorithms, and propose two variants of the well-known KMeans algorithm that can improve their performance with limited labeled data.

We then focus on the problem of semi-supervised clustering with constraints and show how this problem can be studied in the framework of a well-defined probabilistic generative model of a Hidden Markov Random Field. We derive an efficient KMeans-type iterative algorithm, HMRF-KMeans, for optimizing a semi-supervised clustering objective function defined on the HMRF model. We also give convergence guarantees of our algorithm for a large class of clustering distortion measures (e.g., squared Euclidean distance, KL divergence, and cosine distance).

Finally, we develop an active learning algorithm for acquiring maximally informative pairwise constraints in an interactive query-driven framework, which to our knowledge is the first active learning algorithm for semi-supervised clustering with constraints.

Other interesting problems of semi-supervised clustering that we discuss in this thesis include (1) semi-supervised graph-based clustering using kernels, (2) using prior knowledge to improve overlapping clustering of data, (3) integration of both constraint-based and distance-based semi-supervised clustering methods using the HMRF model, and (4) model selection techniques that use the available supervision to automatically select the right number of clusters.

Contents

List of Figures

Chapter 1

Introduction

Two of the most widely-used methods in machine learning for prediction and data analysis are classification and clustering (Duda, Hart, & Stork, 2001; Mitchell, 1997). Classification is a purely supervised learning model, whereas clustering is completely unsupervised. Recently, there has been a lot of interest in the continuum between completely supervised and unsupervised learning (Nigam, 2001; Ghani, Jones, & Rosenberg, 2003). In this chapter, we will give an overview of traditional supervised classification and unsupervised clustering, and then describe learning in the continuum between these two, where we have partially supervised data. We conclude this chapter with a discussion of the thesis contributions.

1.1 Classification

Classification is a supervised task, where supervision is provided in the form of a set of labeled training data, each data point having a class label selected from a fixed set of classes (Mitchell, 1997). The goal in classification is to learn a function from the training data that gives the best prediction of the class label of unseen (test) data points. Generative models for classification learn the joint distribution of the data and class variables by assuming a particular parametric form of the underlying distribution that generated the data points in each class. Subsequently, Bayes Rule is applied to obtain class conditional probabilities that are used to predict the class labels for test points (with unknown class labels) drawn from the same distribution (Ng & Jordan, 2002). In the discriminative framework, the focus is on learning the discriminant function for the class boundaries or a posterior probability for the class labels directly without learning the underlying generative densities (Jaakkola & Haussler, 1999). It can be shown that the discriminative model of classification has better generalization error than the generative model under certain assumptions (Vapnik, 1998), which has made discriminative classifiers, e.g., support vector machines (Vapnik, 1998) and nearest neighbor classifiers (Devroye, Gyorfi, & Lugosi, 1996), very popular for the classification task.

1.2 Clustering

Clustering is an unsupervised learning problem, which tries to group a set of points into clusters such that points in the same cluster are more similar to each other than points in different clusters, under a particular clustering distortion or distance measure (Jain & Dubes, 1988). Here, the learning algorithm just observes a set of points without observing any corresponding class/category labels. Clustering problems can also be categorized as generative or discriminative. In the generative clustering model, a parametric form of data generation is assumed, and the goal in the maximum likelihood formulation is to find the parameters that maximize the probability (likelihood) of generation of the data given the model. In the most general formulation, the number of clusters k is also considered to be an unknown parameter. Such a clustering formulation is called a "model selection" framework, since it has to choose the best value of k under which the clustering model fits the data. We will be assuming that k is known in the clustering frameworks that we will be considering, unless explicitly mentioned otherwise. In the discriminative clustering setting (e.g., graph-theoretic clustering), the clustering algorithm tries to cluster the data so as to maximize within-cluster similarity and minimize between-cluster similarity, based on a similarity matrix defined over the input data set – in this paradigm, it is not necessary to consider an underlying parametric data generation model. In both the generative and discriminative models, clustering algorithms are generally posed as optimization problems and solved by iterative methods like EM (Dempster, Laird, & Rubin, 1977), approximation algorithms like KMedian (Jain & Vazirani, 2001), or heuristic methods like Metis (Karypis & Kumar, 1998).

1.3 Semi-supervised learning

In many practical learning domains (e.g. text processing, bioinformatics), there is a large supply of unlabeled data but limited labeled data, and in most cases it can be expensive to generate that labeled data. Consequently, *semi-supervised learning*, learning from a combination of both labeled and unlabeled data, has become a topic of significant recent interest. The framework of semi-supervised learning is applicable to both classification and clustering.

1.3.1 Semi-supervised classification

Supervised classification has a fixed known set of categories, and category-labeled training data is used to induce a classification function. In this setting, the training can also exploit additional unlabeled data, frequently resulting in a more accurate classification function. Several semi-supervised classification algorithms that use unlabeled data to improve classification accuracy have become popular in the past few years, which include co-training (Blum & Mitchell, 1998), transductive support vector machines (Joachims, 1999), and using Expectation Maximization to incorporate unlabeled data into training (Ghahramani & Jordan, 1994; Nigam, McCallum, Thrun, & Mitchell, 2000). Unlabeled data have also been used to learn good distance measures in the classification setting (Hastie & Tibshirani, 1996). A good review of semi-supervised classification methods is given in (Seeger, 2000).

1.3.2 Semi-supervised clustering

Semi-supervised clustering, which uses class labels or pairwise constraints on some examples to aid unsupervised clustering, has been the focus of several recent projects (Basu, Banerjee, & Mooney, 2002; Klein, Kamvar, & Manning, 2002; Wagstaff, Cardie, Rogers, & Schroedl, 2001; Xing, Ng, Jordan, & Russell, 2003). If the supervised data is available in the form of category labels and the labeled data represent all the relevant categories, then both semi-supervised clustering and semi-supervised classification algorithms can be used for data categorization. However in many domains, knowledge of the relevant categories is incomplete. Unlike semi-supervised classification, semi-supervised clustering (in the model-selection framework) can group data using the categories in the initial labeled data as well as extend and modify the existing set of categories as needed to reflect other regularities in the data.

Existing methods for semi-supervised clustering fall into two general approaches that we call *constraint-based* and *distance-based* methods.

Constraint-based methods

In constraint-based approaches, the clustering algorithm itself is modified so that the available labels or constraints are used to bias the search for an appropriate clustering of the data. The labeled data specify the categories to which an instance belongs, while the pairwise constraints specify whether two instances should be in the same cluster (must-link) or in different clusters (cannot-link). Constraint-based semi-supervised clustering has been done using several techniques, e.g., modifying the clustering objective function so that it includes a term for satisfying specified constraints (Demiriz, Bennett, & Embrechts, 1999), doing clustering using side-information from conditional distributions in an auxiliary space (Sinkkonen & Kaski, 2000), enforcing constraints to be satisfied during the cluster assignment in the clustering process (Wagstaff et al., 2001), and initializing clusters and inferring clustering constraints based on neighborhoods derived from labeled examples (Basu et al., 2002). Constraint-based clustering techniques have been an active topic of research, where recent techniques include variational techniques for constrained clustering using a graphical model (Hiu, Law, Topchy, & Jain, 2005), and feasibility studies for clustering under different types of constraints (Davidson & Ravi, 2005).

Distance-based methods

In distance-based approaches, an existing clustering algorithm that uses a distance measure is employed; however, the distance measure is first trained to satisfy the labels or constraints in the supervised data. Several distance measures have been used for distance-based semi-supervised clustering, including string-edit distance trained using EM (Bilenko & Mooney, 2003), Jensen-Shannon divergence trained using gradient descent (Cohn, Caruana, & McCallum, 2003), Euclidean distance modified by a shortest-path algorithm (Klein et al., 2002), or Mahalanobis distances trained using convex optimization (Bar-Hillel, Hertz, Shental, & Weinshall, 2003; Xing et al., 2003). Several clustering algorithms using trained distance measures have been employed for semi-supervised clustering, including single-link (Bilenko & Mooney, 2003) and complete-link (Klein et al., 2002) agglomerative clustering, EM (Cohn et al., 2003; Bar-Hillel et al., 2003), and KMeans (Bar-Hillel et al., 2003; Xing et al., 2003). Recent techniques in distance-metric learning for clustering include learning a margin-based clustering distortion measure using boosting (Hertz, Bar-Hillel, & Weinshall, 2004), and learning a distance metric transformation that is globally linear but locally non-linear (Chang & Yeung, 2004).

1.4 Thesis contributions

The goal of this research is studying probabilistic models for semi-supervised clustering, deriving algorithms based on these models and subsequently performing detailed experiments to show the effectiveness of these algorithms on different domains. The contributions of this thesis are outlined below:

- We show how supervision in the form of labeled data points can be incorporated into partitional clustering using a well-defined EM framework in Chapter 3.

- We develop a probabilistic generative Hidden Markov Random Field (HMRF) model for semi-supervised clustering with constraints, which is able to perform semi-supervised clustering with a broad class of clustering distance measures, namely Bregman divergences (e.g., squared Euclidean distance, KL divergence) and directional distances (e.g., cosine distance, Pearson's correlation). The HMRF model and the algorithm HMRF-KMEANS that we derive from this model is described in detail in Chapter 4.

- We propose an active learning algorithm for selecting informative constraints in the pairwise constrained semi-supervised clustering model. To our knowledge it is the first active learning algorithm for constraint acquisition in a semi-supervised clustering setting, and it is described in detail in Chapter 5.

- We empirically evaluate the effectiveness of our semi-supervised clustering algorithms by detailed experiments on different domains, both low-dimensional (e.g., handwritten character recognition data sets) and high-dimensional (e.g., text documents). Our experiments conclusively demonstrate that using either labeled supervision or pairwise constraints substantially improve the clustering accuracy on different domains, and that our active learning algorithm is able to acquire informative constraints very effectively.

- We discuss other interesting problems of semi-supervised clustering in Chapter 7, namely (1) integration of both constraint-based and distance-based semi-supervised clustering methods using the HMRF model, (2) semi-supervised graph-based clustering using kernels, (3) using prior knowledge to improve overlapping clustering of data, and (4) model selection techniques that use the available supervision to automatically select the right number of clusters. Finally, Chapter 8 discusses possible extensions of the research presented in this thesis and outlines promising areas of future work in semi-supervised clustering.

Apart from the chapters mentioned above, the thesis also describes related research in the field of semi-supervised clustering in Chapter 6, and finally Chapter 9 concludes the thesis. We begin the thesis by giving some relevant background on clustering in Chapter 2.

Chapter 2

Background

This chapter gives a brief review of clustering algorithms on which our proposed semi-supervised clustering techniques will be applied. It also describes the clustering evaluation measures we will be using in our experiments, and gives an overview of the pre-processing steps we use for text document clustering.

2.1 Notation

A brief review on the notation that we will use in this chapter and the rest of the thesis: $\mathbb{R}^d$ denotes the d-dimensional real vector space; p denotes a probability density function; $X = \{x_i\}_{i=1}^n$ denotes the set of n data points, where the i^{th} data point is a vector represented by x_i whose j^{th} component is x_{ij}; Y denotes the set of n cluster labels, where y_i is the cluster label of the i^{th} data point x_i; other lowercase letters are scalars, e.g., k denotes the number of clusters.

2.2 Overview of clustering algorithms

As explained in Chapter 1, clustering algorithms can be classified into two models — generative or discriminative. There are other categorizations of clustering, e.g., hierarchical or partitional (Jain, Murty, & Flynn, 1999), depending on whether the algorithm clusters the data into a hierarchical structure or generates a flat partitioning of the data.

2.2.1 Hierarchical clustering

In hierarchical clustering, the data is not partitioned into clusters in a single step. Instead, a series of partitions are created, which may run from a single cluster containing all objects to *n* clusters each containing a single object. This gives rise to a hierarchy of clusterings, also known as the cluster dendrogram. Hierarchical clustering methods can be further subdivided into:

- Divisive methods: Create the cluster dendrogram in a top-down divisive fashion, starting with every data point in one cluster and splitting clusters successively according to some measure till a convergence criterion is reached, e.g., Cobweb (Fisher, 1987), recursive cluster-splitting using a statistical transformation (Dubnov, El-Yaniv, Gdalyahu, Schneidman, Tishby, & Yona, 2002), and PDDP or principal direction divisive partitioning (Boley, 1998);

- Agglomerative methods: Create the cluster dendrogram in a bottom-up agglomerative fashion, starting with each data point in its own cluster and merging clusters successively according to a similarity measure till a convergence criterion is reached, e.g., hierarchical agglomerative clustering (Kaufman & Rousseeuw, 1990), Birch (Zhang, Ramakrishnan, & Livny, 1996), etc.

To illustrate hierarchical clustering, let us consider hierarchical agglomerative clustering (HAC) in more detail.

Hierarchical agglomerative clustering

Hierarchical agglomerative clustering (HAC) is a bottom-up hierarchical clustering algorithm. In HAC, points are initially allocated to singleton clusters, and at each step the "closest" pair of clusters are merged, where closeness is defined according to a similarity measure between clusters. The algorithm generally terminates when a specified *convergence criterion* is reached. Different cluster-level similarity measures are used to determine the closeness between clusters to be merged – single-link, complete-link, or group-average (Manning & Schütze, 1999).

Various HAC schemes have been recently shown to have well-defined underlying generative models – single-link HAC corresponds to the probabilistic model of a mixture of branching random walks, complete-link HAC corresponds to uniform equal-radius hyperspheres, whereas group-average HAC corresponds to equal-variance configurations (Kamvar, Klein, & Manning, 2002). The pseudo-code for HAC is given in Fig. 2.1.

```
Algorithm: HIERARCHICAL AGGLOMERATIVE CLUSTERING

Input: Set of data points $X = \{x_i\}_{i=1}^{n}, x_i \in \mathbb{R}^d$

Output: Dendogram representing hierarchical clustering of $X$

Method:

1. Initialize clusters: Each data point $x_i$ is placed in its own cluster $C_i$. These clusters
        form the leaves of the dendogram, and constitute the set of current clusters.

2. Repeat until convergence

2a.    Merge the two closest clusters $C_i$ and $C_j$ from current clusters to get cluster $C$

2b.    Remove $C_i$ and $C_j$ from current clusters, add cluster $C$ to current clusters

2c.    Add parent links from $C_i$ and $C_j$ to $C$ in the cluster dendogram
```

Figure 2.1: HAC algorithm

2.2.2 Partitional clustering

Let $X = \{x_i\}_{i=1}^{n}$, $x_i \in \mathbb{R}^d$, be the set of n data points we want to cluster. A partitional clustering algorithm generates a k-partitioning[1] of the data (k given as input to the algorithm) by grouping the associated data points into k clusters. Partitional algorithms can be classified into the following categories:

- Graph-theoretic: These are discriminative clustering approaches, where an undirected graph $G = (V, E)$ is constructed from the data set, each vertex in $v_i \in V$ corresponding to a data point x_i and the weight of each edge $e_{ij} \in E$ corresponding to the similarity between the data points x_i and x_j according to a domain-specific similarity measure. The k clustering problem becomes equivalent to finding the k-mincut in this graph, which is known to be a NP-complete problem for $k \geq 3$ (Garey & Johnson, 1979). One class of methods for solving the graph partitioning problem take a real relaxation of the NP-complete discrete partitioning problem: these include spectral methods that perform clustering by using the second eigenvector of the graph Laplacian to define a cut (Ng, Jordan, & Weiss, 2001). The other class of methods use heuristics to find low-cost cuts in G: methods like Rock (Guha, Rastogi, & Shim, 1999) and Chameleon (Karypis, Han, & Kumar, 1999) group nodes based on the idea of defining neighborhoods using inter-connectivity of nodes in G, Metis (Karypis & Kumar, 1998) performs fast multi-level heuristics on G at multiple resolutions to

[1] k disjoint subsets $\{X_h\}_{h=1}^{k}$ of X, whose union is X

give good partitions, while Opossum (Strehl & Ghosh, 2000) uses a modified cut criterion to ensure that the resulting clusters are well-balanced according to a specified balancing criterion.

- Density-based: These methods model clusters as dense regions and use different heuristics to find arbitrary-shaped high-density regions in the input data space and group points accordingly. Well-known methods include Denclue, which tries to analytically model the overall density around a point (Hinneburg & Keim, 1998), and WaveCluster, which uses wavelet-transform to find high-density regions (Sheikholesami, Chatterjee, & Zhang, 1998). Density-based methods typically have difficulty scaling up to very high dimensional data ($>$ 10000 dimensions), which are common in domains like text.

- Mixture-model based: In mixture-model based clustering, the underlying assumption is that each of the n data points $\{x_i\}_{i=1}^n$ to be clustered are generated by one of k probability distributions $\{p_h\}_{h=1}^k$, where each distribution p_h is the conditional distribution corresponding to the cluster X_h. The probability of observing any point x_i is given by:

$$\mathrm{P}(x_i|\Theta) = \sum_{i=1}^{k} \alpha_h p_h(x_i|\theta_h)$$

  where $\Theta = (\alpha_1, \ldots, \alpha_k, \theta_1, \ldots, \theta_k)$ is the parameter vector, α_h are the prior probabilities of the clusters ($\sum_{h=1}^k \alpha_h = 1$), and p_h is the probability distribution of cluster X_h parameterized by θ_h. The data generation process is assumed to be as follows –

first, one of the k components is chosen following their prior probability distribution $\{\alpha_h\}_{i=1}^{k}$; then, a data point is sampled following the distribution p_h of the chosen component.

Since the cluster assignment of the points are not known, we assume the existence of a random variable Y that encodes the cluster assignment y_i for each data point x_i and takes values in $\{h\}_{h=1}^{k}$. The goal of clustering in this model is to find the estimates of the parameter vector Θ and the cluster assignment variable Y such that the complete log-likelihood of the data:

$$\mathcal{L}(X,Y|\Theta) = \sum_{i=i}^{n} \log \mathrm{P}(x_i, y_i|\Theta)$$

is maximized, where the i.i.d. (identically and independently distributed) assumption over the data points in X leads to the factoring of the likelihood over the whole data set X into individual probabilities over each data point x_i. Since Y is unknown, the log-likelihood cannot be maximized directly. So, traditional approaches iteratively maximize the *expected* log-likelihood in the Expectation Maximization (EM) framework (Dempster et al., 1977). Starting from an initial estimate of Θ, the EM algorithm iteratively improves the estimates of Θ and $p(Y|X,\Theta)$ such that the expected value of the complete-data log-likelihood is maximized, where the expectation is computed w.r.t. the posterior class distribution $p(Y|X,\Theta)$. It can be shown that the EM algorithm converges to a local maximum of the expected log-likelihood distribution (Dempster et al., 1977), and the final estimates of the conditional distribution $p(Y|X,\Theta)$, on convergence of the algorithm, are used to find the cluster assignments of the points in X.

Most of the work in this area has assumed that the individual mixture density components p_h are Gaussian, and in this case the parameters of the individual Gaussians are estimated by the EM procedure. The popular KMeans clustering algorithm (MacQueen, 1967) can be shown to be an EM algorithm on a mixture of k Gaussians under certain assumptions: details of this derivation are shown in Sec. 2.3.1. Another interesting model for Gaussian mixture model-based clustering is AutoClass (P. Cheeseman & Freeman, 1988), which also has a Bayesian model selection component for choosing the optimal number of clusters.

2.3 Representative clustering algorithm: KMeans

In our thesis, we have chosen KMeans as our representative partitional clustering algorithm on which the proposed semi-supervised schemes will be applied. The following sections give brief descriptions of KMeans and COP-KMeans algorithm, the latter being a recently proposed semi-supervised KMeans algorithm that we will compare our algorithms to.

2.3.1 KMeans

KMeans is a partitional clustering algorithm that performs iterative relocation to partition a data set into k clusters, locally minimizing the overall distortion measure between the data points and the cluster means (a.k.a. centroids). For a set of data points $X = \{x_i\}_{i=1}^n, x_i \in \mathbb{R}^d$, the KMeans algorithm creates a k-partitioning $\{X_h\}_{h=1}^k$ of X so that if $\{\mu_h\}_{h=1}^k$ represent the k partition centroids, then the following objective function

$$\mathcal{J}_{\text{kmeans}} = \sum_{h=1}^{k} \sum_{x_i \in X_h} ||x_i - \mu_h||^2 \tag{2.1}$$

is locally minimized. Lowering this objective function leads to getting tighter clusters, where each point gets closer to its cluster centroid. Note that finding the global optima for the KMeans objective function is an NP-complete problem (Garey, Johnson, & Witsenhausen, 1982). Considering y_i is the cluster assignment of the point x_i, where $y_i \in \{h\}_{h=1}^k$, an equivalent form of the KMeans clustering objective function, which we will be using interchangeably, is:

$$\mathcal{J}_{\text{kmeans}} = \sum_{x_i \in X} ||x_i - \mu_{y_i}||^2 \tag{2.2}$$

The pseudocode for KMeans is given in Fig. 2.2. Note that under certain assumptions, KMeans can be considered as fitting a mixture of Gaussians to a data set, which is described in more detail in Sec. 3.3.1.

If we have the additional constraint that the centroids $\{\mu_h\}_{h=1}^k$ are restricted to be

```
Algorithm: KMEANS

Input: Set of data points $X = \{x_i\}_{i=1}^{n}, x_i \in \mathbb{R}^d$, number of clusters $k$

Output: Disjoint $k$ partitioning $\{X_h\}_{h=1}^{k}$ of $X$ such that KMeans objective
    function is optimized

Method:

1. Initialize clusters: Initial centroids $\{\mu_h^{(0)}\}_{h=1}^{k}$ are selected at random

2. Repeat until convergence

2a.   assign_cluster: Assign each data point $x$ to the cluster $h^*$ (i.e. set $X_{h^*}^{(t+1)}$),
      for $h^* = \arg\min_h \|x - \mu_h^{(t)}\|^2$

2b.   estimate_means: $\mu_h^{(t+1)} \leftarrow \frac{1}{|X_h^{(t+1)}|} \sum_{x \in X_h^{(t+1)}} x$

2c.   $t \leftarrow (t+1)$
```

Figure 2.2: KMeans algorithm

selected from X, then the resulting problem is called KMedian clustering. KMedian clustering corresponds to an integer programming problem, for which many approximation algorithms have been proposed (Jain & Vazirani, 2001; Mettu & Plaxton, 2000).

2.3.2 SP-KMeans

In certain high dimensional data, e.g. text, Euclidean distance is not a good measure of similarity. Certain high dimensional spaces like text have good directional properties, which has made directional similarity measures like L_2 normalized dot product (cosine similarity) between the vector representations of text data a popular measure of similarity in the information retrieval community (Baeza-Yates & Ribeiro-Neto, 1999). Note that other similarity measures, e.g., probabilistic document overlap (Goldszmidt & Sahami, 1998), have also been used successfully for text clustering, but we will be focusing on cosine similarity in our work.

Spherical KMeans (SP-KMeans) is a version of KMeans that uses cosine similarity as its underlying similarity metric. In the SP-KMeans algorithm, standard KMeans is applied to data vectors $\{x_i\}_{i=1}^n$ that have been normalized to have unit L_2 norm, so that the data points lie on a unit sphere (Dhillon & Modha, 2001). Note that in SP-KMeans, the centroid vectors $\{\mu_h\}_{h=1}^k$ are also constrained to lie on the unit sphere. Assuming $\|x_i\| = \|\mu_h\| = 1, \forall i, h$ in Eqn. (2.1), we get $\|x_i - \mu_h\|^2 = 2 - 2x_i^T \mu_h$. Then, the clustering problem can be equivalently formulated as that of maximizing the objective function:

$$\mathcal{J}_{\text{sp-kmeans}} = \sum_{h=1}^{k} \sum_{x_i \in X_h} x_i^T \mu_h \tag{2.3}$$

The centroid μ_h of the h^{th} cluster is the mean of all the points in that cluster, normalized to have unit L_2 norm. The SP-KMeans algorithm gives a local maximum of this objective function. The SP-KMeans algorithm is computationally efficient for sparse high dimensional data vectors, which are very common in domains like text clustering. For this reason, we have used SP-KMeans in our experiments with text data (see Sec. 2.5).

2.3.3 COP-KMeans

In this thesis, we will be comparing some of our proposed semi-supervised KMeans algorithms to another recently proposed semi-supervised variant of KMeans, called COP-KMeans (Wagstaff et al., 2001). In COP-KMeans, initial background knowledge, provided in the form of constraints between instances in the data set, is used in the clustering process. It uses two types of constraints, *must-link* (two instances have to be together in the same cluster) and *cannot-link* (two instances have to be in different clusters).

In the initialization step, COP-KMeans chooses cluster centers randomly; but as each one is chosen, any must-link constraints that it participates in are enforced, i.e., all items that the chosen instance must link to are assigned to the new cluster, so that they cannot later be chosen as the center of another cluster. After cluster initialization, COP-

KMeans iterates between the following 2 steps till the pre-defined convergence criterion is reached:

- `assign_cluster`: Assign each data point to the closest cluster such that no must-link or cannot-link constraint is violated by the assignment. If no such assignment exists, the algorithm **aborts**;
- `estimate_means`: Update each cluster centroid to be the average of all the points assigned to that cluster.

Note that the COP-KMeans algorithm is not robust to inconsistencies in potentially noisy constraints, since in that case the algorithm does not find a consistent assignment and aborts in the cluster assignment step.

2.4 Clustering evaluation measures

Evaluation of the quality of output of clustering algorithms is a difficult problem in general, since there is no "gold-standard" solution in clustering. The commonly used clustering validation measures can be categorized as *internal* or *external*. Internal validation measures, e.g., the ratio of the average inter-cluster to intra-cluster distance (the lower the better), need only the data and the clustering for their measurement. External validation measures, on the other hand, match the clustering solution to some known prior knowledge, e.g., an underlying class labeling of the data. Many data sets in supervised learning have class information: we can evaluate a clustering algorithm by applying it to such a data set (with the class label information removed), and then using the class labels of the data as the gold standard against which we can compare the quality of the data clustering obtained.

In our experiments, we have used three metrics for cluster evaluation: *normalized mutual information* (NMI), *pairwise F-measure*, and *objective function*. Of these, normalized mutual information and pairwise F-measure are external clustering validation metrics that estimate the quality of the clustering with respect to a given underlying class labeling of the data.

For clustering algorithms which optimize a particular objective function, we can report the value of the objective function when the algorithm converges. For KMeans and SP-KMeans, the objective function values reported are $\mathcal{J}_{\text{kmeans}}$ from Eqn. (2.1) and $\mathcal{J}_{\text{sp-kmeans}}$ from Eqn. (2.3). For the semi-supervised versions of KMeans, we report their corresponding objective function values, e.g., $\mathcal{J}_{\text{hmrf-kmeans}}$ from Eqn. (4.10) for HMRF-KMEANS. Since all the semi-supervised clustering algorithms we propose are iterative methods that locally minimize the corresponding clustering objective functions, looking at the objective function value after convergence would give us an idea of whether the semi-supervised algorithm under consideration generated a good clustering that converged to a good local optimum of the objective function.

One external clustering evaluation measure is normalized mutual information (NMI), which determines the amount of statistical information shared by the random variables representing the cluster assignments and the pre-labeled class assignments of the data points. We compute NMI following the methodology of Strehl et al. (Strehl, Ghosh, & Mooney, 2000). NMI measures how closely the clustering algorithm could reconstruct the underlying label distribution in the data. If C is the random variable denoting the cluster assignments of the points, and K is the random variable denoting the underlying class labels on the points then the NMI measure is defined as:

$$NMI = \frac{I(C;K)}{(H(C)+H(K))/2} \tag{2.4}$$

where $I(X;Y) = H(X) - H(X|Y)$ is the mutual information between the random variables X and Y, $H(X)$ is the Shannon entropy of X, and $H(X|Y)$ is the conditional entropy of X given Y (Cover & Thomas, 1991). For a discrete random variable X, $H(X) = -\sum_{x \in X} p(x) \log p(x)$ and $H(X|Y) = -\sum_{x \in X} p(x|Y) \log p(x|Y)$, where $p(x)$ and $p(x|Y)$ are respectively the probability of X and the conditional probability of X given Y. The normalization by the average entropy of C and K makes the value of NMI stay between 0 and 1.

Pairwise F-measure is defined as the harmonic mean of pairwise precision and recall, where the traditional information retrieval measures are adapted for evaluating clustering by considering pairs of points. For any pair of points, the decision to cluster this pair into same or different clusters is considered to be correct if it matches with the underlying class labeling available for the points (Bilenko & Mooney, 2002). Pairwise F-measure is defined as:

$$\text{Precision} = \frac{\text{Number of pairs correctly predicted in same cluster}}{\text{Total number of pairs predicted in same cluster}}$$

$$\text{Recall} = \frac{\text{Number of pairs correctly predicted in same cluster}}{\text{Total number of pairs actually in same cluster}}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision + Recall}} \tag{2.5}$$

Pairwise F-measure is related to measures like Rand Index (Klein et al., 2002; Wagstaff et al., 2001; Xing et al., 2003) that have been used in other semi-supervised clustering research. NMI has also become a popular clustering evaluation metric (Banerjee, Dhillon, Ghosh, & Sra, 2003; Dom, 2001; Fern & Brodley, 2003). Recently, a symmetric cluster evaluation metric based on mutual information has been proposed, which has some useful properties, e.g., it is a true metric in the space of clusterings (Meila, 2003). In most of our experiments, the comparative results of different algorithms, using NMI and pairwise F-measure, were qualitatively similar.

Note that the external cluster validation measures we have used (e.g., pairwise F-measure and NMI) are not completely definitive, since the clustering can find a grouping of the data that is different from the underlying class structure. For example, in our initial experiments on clustering articles from the CMU 20 Newsgroups data (where the main Usenet newsgroup to which an article was posted is considered to be its class label), we found one cluster that had articles from four underlying classes — `alt.atheism`, soc.religion.christian, talk.politics.misc, and talk.politics.guns. On closer observation, we noticed that all the articles in the cluster were about the David Koresh episode; this is a valid cluster, albeit different from the grouping suggested by the underlying class labels.

If we had human judges to evaluate the cluster quality, we could find an alternate external cluster validation measure — we could ask the human judges to rank data categorizations generated by humans and the clustering algorithm, and the quality of a clustering output would be considered to be high if the human judges could not reliably discriminate between a human categorization of the data and the grouping generated by the clustering algorithm. Since this is a time- and resource-consuming method of evaluation in the academic setting, we have used automatic external cluster validation methods like pairwise F-measure and NMI in our experiments.

2.5 Pre-processing of text documents for clustering

In our experiments with text documents we used the vector space model, where a text document is represented as a sparse high-dimensional vector of weighted term counts (Salton & McGill, 1983). The creation of the vector space model can be divided into two stages. At first, the content-bearing terms (which are typically words or short phrases) are extracted from the document text and the weight of each term in the document vector is set to the count of the corresponding term in the document. In the second stage, the terms are suitably weighted according to information retrieval principles to increase the weights of important terms.

Some terms in a document do not describe any important content, e.g., common words like “the”, “is” – these words are called stop-words. While processing a document to count the number of occurrences of each term and create the term count vector in the first phase, these stop-words are filtered from the document and not included in the vector. Note that this vector is often more than 99% sparse, since the dimensionality of the vector is equal to the number of terms in the whole document collection but most documents just have a small subset of these terms. In our experiments, we used the MC toolkit[2] for creating the document vectors from raw text documents.

In the second phase, the term-frequencies or counts of the terms are multiplied by the inverse document frequency of a term in the document collection. This is done so that terms that are common to most documents in a document collection (e.g., “god” is a common term in a collection of articles posted to newsgroups like `alt.atheism` or `soc.religion.christian`) are given lesser weight, since they are not very content-bearing in the context of the collection. This method of term weighting, called “Term Frequency and Inverse Document Frequency” (TFIDF), is a popular method of pre-processing documents in the information retrieval community (Baeza-Yates & Ribeiro-Neto, 1999).

The TFIDF weighting procedure we use is as follows. If f_{ij} is the frequency of the i^{th} term in the j^{th} document, then the corresponding term frequency (TF) tf_{ij} is f_{ij}

[2] http://www.cs.utexas.edu/users/jfan/dm

normalized across the entire document corpus:

$$tf_{ij} = f_{ij}$$

The inverse document frequency (IDF) idf_i of the i^{th} term is defined as:

$$idf_i = \log_2(N/df_i)$$

where N is the total number of documents in the corpus and df_i is the total number of documents containing the i^{th} term. The overall TFIDF score w_{ij} of the i^{th} term in the j^{th} document is therefore:

$$w_{ij} = tf_{ij} idf_i = f_{ij} \log_2(N/df_i)$$

After TFIDF processing, terms which have a very low (occurring in less than 5 documents) and very high frequency (occurring in more than 95% of the documents) are also removed from the documents (Dhillon, Fan, & Guan, 2001). Finally, the weights of the document vectors are re-normalized so that every document has unit length according to the L_2 norm. While clustering, similarity between two documents can now be computed using the dot product between the document vectors, which would give the cosine similarity between the vector representations of the documents. The similarity of documents d_{j1} and

d_{j2} are computed as follows:

$$sim(d_{j1}, d_{j2}) = \sum_{i=1}^{|V|} w'_{ij1} w'_{ij2}$$

where $|V|$ is the size of the term vocabulary and w' represents the TFIDF weights after renormalization. In practice, this sum calculation can be performed very efficiently by using sparse representations of document vectors and computing the sum only over the terms in the shorter document.

Some other specific pre-processing steps were also performed based on the types of the documents, e.g., headers and email signatures were removed for newsgroup articles, and HTML tags were removed for webpages.

Chapter 3

Semi-supervised Clustering with Labels

This chapter describes how supervision in the form of labeled data can be incorporated into clustering (Basu et al., 2002). We use the labeled data to generate seed clusters that initialize a clustering algorithm, and use constraints generated from the labeled data to guide the clustering process. The underlying intuition is that proper seeding biases clustering towards a good region of the search space, thereby reducing the chances of it getting stuck in poor local optima while simultaneously producing a clustering similar to the specified labels.

3.1 Problem definition

Given a data set X, as previously mentioned, KMeans clustering of the data set generates a k-partitioning $\{X_h\}_{h=1}^{k}$ of X so that the KMeans objective is locally minimized. Let $S \subseteq X$, called the *seed set*, be the subset of data-points on which supervision is provided as follows: for each $x_i \in S$, we have the label h of the partition X_h to which it belongs. We assume that corresponding to each partition X_h of X, there is at least one seedpoint $x_i \in S$ (we will relax this assumption for our experiments with incomplete seeding). Note that we get a disjoint k-partitioning $\{S_h\}_{h=1}^{k}$ of the seed set S, so that all $x_i \in S_h$ belongs to X_h according to the supervision. This partitioning of the seed set S forms the *seed clustering*. The goal is to guide the KMeans algorithm towards the desired clustering of the whole data as illustrated by the seed clustering.

3.2 SEEDED-KMEANS and CONSTRAINED-KMEANS algorithms

We propose two algorithms for semi-supervised clustering with labeled data: SEEDED-KMEANS and CONSTRAINED-KMEANS.

In SEEDED-KMEANS, the seed clustering is used to initialize the KMeans algorithm. Thus, rather than initializing KMeans from k random means, the centroid of the h^{th} cluster is initialized with the centroid of the h^{th} partition S_h of the seed set. The seed clustering is only used for initialization, and the seeds are not used in the following steps of the algorithm. The algorithm is presented in detail in Fig. 3.1. In CONSTRAINED-KMEANS,

```
Algorithm: SEEDED-KMEANS
Input: Set of data points $X = \{x_i\}_{i=1}^{n}, x_i \in \mathbb{R}^d$, number of clusters $k$, set
    $S = \cup_{h=1}^{k} S_h$ of initial seeds
Output: Disjoint $k$ partitioning $\{X_h\}_{h=1}^{k}$ of $X$ such that KMeans objective
    function is optimized
Method:
1. Initialize clusters: $\mu_h^{(0)} \leftarrow \frac{1}{|S_h|} \sum_{x \in S_h} x$, for $h = 1, \dots, k; t \leftarrow 0$
2. Repeat until convergence
2a.    assign_cluster: Assign each data point $x$ to the cluster $h^*$ (i.e. set $X_{h^*}^{(t+1)}$),
       for $h^* = \underset{h \in \{1,\dots,k\}}{\arg\min} \|x - \mu_h^{(t)}\|^2$
2b.    estimate_means: $\mu_h^{(t+1)} \leftarrow \frac{1}{|X_h^{(t+1)}|} \sum_{x \in X_h^{(t+1)}} x$
2c.    $t \leftarrow (t+1)$
```

Figure 3.1: Seeded-KMeans algorithm

the seed clustering is used to initialize the KMeans algorithm as described for the SEEDED-KMEANS algorithm. However, in the subsequent steps, the cluster memberships of the data points in the seed set are not re-computed in the `assign_cluster` step of the algorithm – the cluster labels of the seed data are kept unchanged, and only the labels of the non-seed data are re-estimated. The algorithm is given in detail in Fig. 3.2. CONSTRAINED-KMEANS seeds the KMeans algorithm with the given labeled data and keeps that labeling unchanged throughout the algorithm. In SEEDED-KMEANS, the given labeling of the seed data may be

```
Algorithm: CONSTRAINED-KMEANS
Input: Set of data points $X = \{x_i\}_{i=1}^{n}, x_i \in \mathbb{R}^d$, number of clusters $k$, set
    $S = \cup_{h=1}^{k} S_h$ of initial seeds
Output: Disjoint $k$ partitioning $\{X_h\}_{h=1}^{k}$ of $X$ such that the KMeans objective
    function is optimized
Method:
1. Initialize clusters: $\mu_h^{(0)} \leftarrow \frac{1}{|S_h|} \sum_{x \in S_h} x$, for $h = 1, \ldots, k; t \leftarrow 0$
2. Repeat until convergence
2a.   assign_cluster: For $x \in S$, if $x \in \mathcal{S}_h$ assign $x$ to the cluster $h$ (i.e., set $X_h^{(t+1)}$).
      For $x \notin S$, assign $x$ to the cluster $h^*$ (i.e. set $X_{h^*}^{(t+1)}$), for $h^* = \underset{h \in \{1,\ldots,k\}}{\arg\min} \|x - \mu_h^{(t)}\|^2$
2b.   estimate_means: $\mu_h^{(t+1)} \leftarrow \frac{1}{|X_h^{(t+1)}|} \sum_{x \in X_h^{(t+1)}} x$
2c.   $t \leftarrow (t+1)$
```

Figure 3.2: Constrained-KMeans algorithm

changed in the course of the algorithm. CONSTRAINED-KMEANS is appropriate when the initial seed labeling is noise-free, or if the user does not want the labels of the seed data to change. On the other hand, SEEDED-KMEANS is more appropriate in the presence of noisy seeds, since it does not enforce the seed labels to remain unchanged during the clustering iterations and can therefore abandon noisy seed labels after the initialization step.

3.3 Underlying probabilistic motivation

The two proposed semi-supervised KMeans algorithms, SEEDED-KMEANS and CONSTRAINED-KMEANS, can be motivated by considering KMeans in the EM framework, as shown in the following section.

3.3.1 Interpretation of KMeans as EM

Both KMeans and SP-KMeans are model-based clustering algorithms, having well-defined underlying generative models. As mentioned earlier, KMeans can be considered as fitting a mixture of Gaussians to a data set under certain assumptions. The assumptions are that the prior distribution $\{\alpha_h\}_{h=1}^k$ of the Gaussians is uniform, i.e., $\alpha_h = 1/k, \forall h$, and that each Gaussian has identity covariance. Then, the parameter set Θ in the EM framework consists of just the k means $\{\mu_h\}_{h=1}^k$. With these assumptions, one can show that (Bilmes, 1997):

$$\begin{aligned} \mathbf{E}_{Y|X,\Theta}[\log \mathrm{P}(X,Y|\Theta)] &= \sum_{h=1}^{k}\sum_{i=1}^{n} \log(\alpha_h \cdot \frac{1}{(2\pi)^{d/2}} e^{-\|x_i-\mu_h\|^2})\, p(y_h|x_i,\Theta) \qquad (3.1) \\ &= -\sum_{h=1}^{k}\sum_{i=1}^{n} \|x_i-\mu_h\|^2\, p(y_h|x_i,\Theta) + c, \end{aligned}$$

where c is a constant and $(Y = h)$ is denoted by y_h. Further assuming that

$$p(y_h|x_i,\Theta) = \begin{cases} 1 & \text{if } h = \arg\min_{l} \|x_i-\mu_l\|^2, \\ 0 & \text{otherwise,} \end{cases} \qquad (3.2)$$

and replacing it in Eqn. (3.2), we note that the expectation term comes out to be the negative of the well-known KMeans objective function with an additive constant.[1] Thus, the problem of maximizing the expected log-likelihood under these assumptions is same as that of minimizing the KMeans objective function. Keeping in mind the assumption in Eqn. (3.2), the KMeans objective can be written as

$$\mathcal{J}_{\text{kmeans}} = \sum_{h=1}^{k} \sum_{i=1}^{n} \|x_i - \mu_h\|^2 \, p(y_h|x_i, \mu_h). \tag{3.3}$$

In a similar fashion, SP-KMeans can be considered as fitting a mixture of von Mises-Fisher distributions to a data set under some assumptions (Banerjee et al., 2003). Note that in the SP-KMeans framework (Sec. 2.3.2), since every point lies on the unit sphere so that $\|x_i\| = \|\mu_h\| = 1$, the expectation term in Eqn. (3.2) becomes equivalent to

$$\mathbf{E}_{Y|X,\Theta}[\log p(X,Y|\Theta)] = \sum_{h=1}^{k} \sum_{i=1}^{n} x_i^T \mu_h \, p(y_h|x_i, \Theta) + c.$$

So, maximizing the SP-KMeans objective function is equivalent to maximizing the expectation of the complete-data log-likelihood in the E-step of the EM algorithm.

[1] The assumption in Eqn. (3.2) can also be derived by assuming the covariance of the Gaussians to be $\varepsilon\mathbf{I}$ and letting $\varepsilon \to 0^+$ (Kearns, Mansour, & Ng, 1997).

3.3.2 Discussion of SEEDED-KMEANS and CONSTRAINED-KMEANS

According to the discussion in the previous section, the only "missing data" for the KMeans problem are the conditional distributions of the cluster labels given the points and the parameters, i.e., $p(y_h|x_i,\mu_h)$. Knowledge of these distributions solves the clustering problem, but normally there is no way to compute it. In the semi-supervised clustering framework, label information is available on some of the data points, which specifies the corresponding conditional distributions. Thus, semi-supervision by providing labeled data is equivalent to providing information about the conditional distributions $p(y_h|x_i,\mu_h)$.

In standard KMeans without any initial supervision, the k means are chosen randomly in the initial M-step and the data-points are assigned to the nearest means in the subsequent E-step. As explained above, every point x_i in the data set has k possible conditional distributions associated with it (each satisfying Eqn. (3.2)) corresponding to the k means to which it can belong. This assignment of data point x_i to a random cluster in the first E-step is similar to picking one conditional distribution at random from the k possible conditional distributions.

In SEEDED-KMEANS, the initial supervision is equivalent to specifying the conditional distributions $p(y_h|x_i,\mu_h)$ for the seed points $x_i \in \mathcal{S}$. The specified conditional distributions of the seed data are just used in the initial M-step of the algorithm, and $p(y_h|x_i,\mu_h)$ is re-estimated for all $x_i \in X$ in the following E-steps of the algorithm.

In CONSTRAINED-KMEANS, the initial M-step is same as SEEDED-KMEANS.

The difference is that for the seed data points, the initial labels, i.e., the conditional distributions $p(y_h|x_i, \mu_h)$, are kept unchanged throughout the algorithm, whereas the conditional distribution for the non-seed points are re-estimated at every E-step.

It can also be shown that getting good seeding is very essential for centroid-based clustering algorithms like KMeans. As shown in Sec. 2.3.1, under certain generative model-based assumptions, one can connect the mixture of Gaussians model to the KMeans model. A direct calculation using Chernoff bounds shows that if a particular cluster (with an underlying Gaussian model) with true centroid μ is seeded with m points (drawn independently at random from the corresponding Gaussian distribution) and the estimated centroid is $\hat{\mu}$, then

$$\mathrm{P}(|\hat{\mu} - \mu| \geq \delta) \leq e^{-\delta^2 m/2}, \tag{3.4}$$

where $\delta \in \mathbb{R}+$ (Banerjee, 2001). Thus, the probability of deviation of the centroid estimates falls exponentially with the number of seeds, and hence seeding results in good initial centroids.

3.4 Convergence of SEEDED-KMEANS and CONSTRAINED-KMEANS

Theorem: *The* SEEDED-KMEANS *and* CONSTRAINED-KMEANS *algorithms converge to a local minima of* $\mathcal{J}_{kmeans}$.

Proof: The SEEDED-KMEANS and CONSTRAINED-KMEANS algorithms alternate between updating the assignment of points to clusters and updating the cluster centroids. If the individual updates of objective function $\mathcal{J}_{\text{kmeans}}$ in each of these two steps is non-increasing, then after each iteration of SEEDED-KMEANS and CONSTRAINED-KMEANS the objective function in Eqn. (3.3) is guaranteed to be non-increasing. Let us inspect each step in the updates to ensure that this is indeed the case.

In SEEDED-KMEANS, the labeled data are used only for cluster initialization – henceforth, both the cluster assignment and centroid re-estimation steps are same as normal KMeans. Since KMeans is guaranteed to converge to a local minima of the objective function $\mathcal{J}_{\text{kmeans}}$ (Selim & Ismail, 1984), SEEDED-KMEANS also has the same guarantees.

For analyzing CONSTRAINED-KMEANS, let us look at the cluster assignment and centroid re-estimation steps separately. First, let us consider the cluster assignment step: according to Sec. 3.3.1, the cluster assignment step is equivalent to the E-step of the corresponding EM update. Each point x_i moves to a new cluster h only if the following component of $\mathcal{J}_{\text{kmeans}}$, contributed by the point x_i, is decreased with the move:

$$\sum_{h=1}^{k} \|x_i - \mu_h\|^2 \, p(y_h|x_i, \mu_h). \tag{3.5}$$

For points $x_i \notin S$, the cluster assignment minimizes the above contribution of x_i to the objective function $\mathcal{J}_{\text{kmeans}}$. For points $x_i \in S$, the cluster assignment remains unchanged; as a result, the contribution of each $x_i \in S$ to the objective function remains unchanged. So, the cluster assignment step of CONSTRAINED-KMEANS either decreases the overall objective function $\mathcal{J}_{\text{kmeans}}$ or keeps it unchanged.

For analyzing the centroid re-estimation step, let us consider an equivalent form of Eqn. (3.3):

$$\mathcal{J}_{\text{kmeans}} = \sum_{h=1}^{k} \sum_{x_i \in X_h} \|x_i - \mu_h\|^2 \, p(y_h | x_i, \mu_h). \tag{3.6}$$

In the centroid re-estimation step, each cluster centroid μ_h is re-estimated so that Eqn. (3.6) is minimized with respect to the centroids. Taking the derivative of Eqn. (3.6) with respect to μ_h and setting it to zero, the μ_h that minimizes Eqn. (3.6), given the cluster assignments, turns out to be the mean of the points in the partition X_h (which includes both the seed points already in X_h and the non-seed points that were assigned to X_h in the previous cluster assignment step). This minimizes the component of $\mathcal{J}_{\text{kmeans}}$ in Eqn. (3.6) contributed by the partition X_h. So, given the cluster assignments, $\mathcal{J}_{\text{kmeans}}$ will decrease or remain the same in this step. Note that the result of the mean of the points in a cluster being the choice of the centroid that minimizes the objective func the objective function in the M-step of EM holds for both cosine distance (Banerjee et al., 2003) and the general class of regular Bregman divergences (Banerjee, Merugu, Dhillon, & Ghosh, 2004).

Hence the objective function decreases after every cluster assignment and centroid re-estimation step in CONSTRAINED-KMEANS. Now, note that the objective function is bounded below by zero. CONSTRAINED-KMEANS results in a decreasing sequence of objective function values, the value sequence must have an accumulation point. The accumulation point in this case will be a fixed point of $\mathcal{J}_{\text{kmeans}}$ since neither updating the assignments or the parameters can further decrease the value of the objective function. As a result, the CONSTRAINED-KMEANS algorithm will converge to a fixed point (local minimum) of the objective. In practice, convergence can be determined if subsequent iterations of CONSTRAINED-KMEANS result in insignificant changes in $\mathcal{J}_{\text{kmeans}}$. ■

3.5 Experiments

The experimental results presented in this section demonstrate the advantages of SEEDED-KMEANS and CONSTRAINED-KMEANS over standard random seeding and COP-KMeans (Wagstaff et al., 2001), a previously developed semi-supervised clustering algorithm described in Sec. 2.3.3.

We show results of our experiments on both high-dimensional text data sets (Yahoo! News K-series and subsets of CMU 20 Newsgroups), as well as on a low-dimensional data set from the UCI repository (Iris). For each data set, we ran 4 clustering algorithms – SEEDED-KMEANS, CONSTRAINED-KMEANS, COP-KMeans, and random KMeans. In random KMeans, the k means were initialized by taking the mean of the entire data and randomly perturbing it k times (Fayyad, Reina, & Bradley, 1998). This technique of initialization has given good results in unsupervised KMeans in previous work (Dhillon et al., 2001). We compared the performance of these 4 methods on the different data sets with varying seeding and noise levels, using 10-fold cross validation. For the high-dimensional data sets, SP-KMeans was used as the underlying KMeans algorithm for all the 4 KMeans variants, while standard KMeans with squared Euclidean distance was used on the low-dimensional data sets.

3.5.1 Data sets

Iris is a low-dimensional data set from the UCI repository (Blake & Merz, 1998), where the task is to categorize a group of 150 4-dimensional vectors, representing Iris flowers, into 3 species.

Among the high-dimensional data sets, the 20 Newsgroups data set (*20-Newsgroups-1000*) is a collection of 20,000 messages, collected from 20 different Usenet newsgroups – 1000 messages from each of the 20 newsgroups were chosen, and the data set was partitioned by newsgroup name.[2] The text documents were pre-processed using the methodology described in Sec. 2.5, which includes removal of non-content-bearing stop-words, TF-IDF weighting, and removal of very high-frequency and low-frequency words. For the *20-Newsgroups-1000* data set, the vector space model had a vocabulary of 21,631 words. The Yahoo! News K-series (*Yahoo! News*) data set[3] is a collection of 2340 Yahoo! news articles belonging to one of 20 different Yahoo! categories. The vector space model of the K1 set from the Yahoo! K-series has 12,229 words, so that the data-points reside in a 12,229 dimensional space.

We derived subsets from the *20-Newsgroups-1000* collection. From the original data set, we created *20-Newsgroups-100*, a reduced data set having a random subsample of

[2] http://www.ai.mit.edu/people/jrennie/20_newsgroups

[3] ftp://ftp.cs.umn.edu/users/boley/PDDPdata

100 documents from each of the 20 newsgroups in the original data. We created the other data subsets by selecting 3 categories from the original *20-Newsgroups-1000* collection:

- *3-News-Similar-1000* consists of 1000 documents each from 3 newsgroups on similar topics (`comp.graphics`, `comp.os.ms-windows`, `comp.windows.x`). This data subset has 3000 points in a vector space of 5950 words, and the underlying clusters are not well separated due to the similarity between the topics;

- *3-News-Related-1000* consists of 1000 postings each from 3 newsgroups on related topics (`talk.politics.misc`, `talk.politics.guns`, and `talk.politics.mideast`), with overall 3000 documents and 10,091 words;

- *3-News-Different-1000* consists of 1000 articles each from 3 newsgroups that cover different topics (`alt.atheism`, `rec.sport.baseball`, `sci.space`). It has 3000 points in 7670 dimensions, with the clusters being well-separated.

The data set *Small-20 Newsgroups* was created to study the effect of data set size on the clustering performance of the algorithms. We created the 3 subsets, having articles from 3 newsgroups, to study the effect of data separability on the algorithms. For each data set, the clustering algorithms were asked to generate the same number of clusters as the number of underlying classes in the data set.

3.5.2 Methodology

For all the algorithms, we generated learning curves with 10-fold cross-validation on each data set. For studying the effect of seeding, 10% of the data set was set aside as the test set at any particular fold. The training sets at different points of the learning curve were obtained from the remaining 90% of the data by varying the seed fraction from 0.0 to 1.0 in steps of 0.1, and the results at each point on the learning curve were obtained by averaging over 10 folds. The clustering algorithm was run on the whole data set, but we calculated the evaluation metrics only on the test set: this was done to estimate the generalization performance of the semi-supervised clustering algorithm on instances for which no labels were provided. For these experiments we used the clustering objective function and Normalized Mutual Information (NMI), as described in Sec. 2.4, as the evaluation measures. For studying the effects of noise in the seeding, we generated learning curves by keeping a fixed fraction of seeding and varying the noise fraction.

3.5.3 Seed and noise generation

In SEEDED-KMEANS and CONSTRAINED-KMEANS, the seeds at any point on the learning curve were selected from the data set according to the corresponding seed fraction. In COP-KMeans, the must-link and the cannot-link constraints are generated from the specified seeds. The *k* cluster centers are chosen randomly, but as each one is chosen, any must-link constraints that it participates in are enforced, i.e., all items that the chosen instance must link to are assigned to the new cluster, so that they cannot later be chosen as the center of another cluster (Wagstaff et al., 2001).

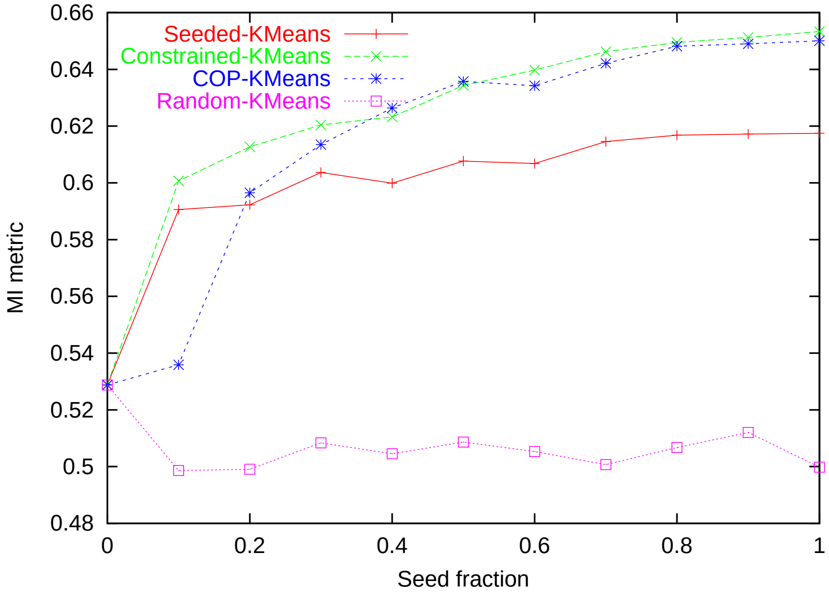

Figure 3.3: Comparison of NMI on *20-Newsgroups-1000* data, noise fraction = 0

In a real-life application, since the semi-supervision will be provided by a human user, there is a chance that the supervision may be erroneous in some cases. We simulate such labeling noise in our experiments by changing the labels of a fraction of the seed examples to a random incorrect value.

3.5.4 Results and discussion

NMI with respect to seeding: For the zero-noise case, the semi-supervised algorithms perform better than the unsupervised algorithm in terms of the NMI measure (Figs. 3.3,3.5,3.7,3.9), irrespective of the size of the data set. CONSTRAINED-KMEANS performs at least as well

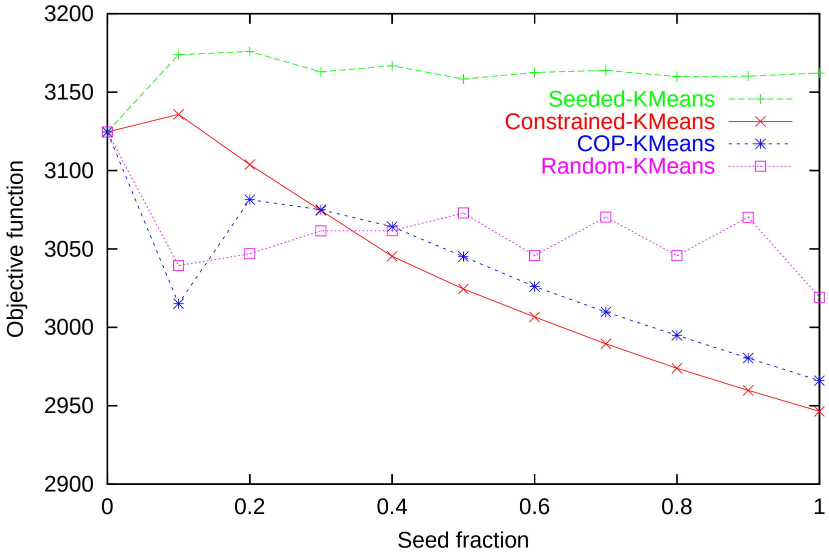

Figure 3.4: Comparison of objective function on *20-Newsgroups-1000* data, noise fraction = 0

as the SEEDED-KMEANS, since the former uses the correct user bias introduced by the user-labeled seeds throughout the execution of the algorithm in the zero-noise case. In spite of being a constrained algorithm, COP-KMeans does not necessarily perform as well as CONSTRAINED-KMEANS, mainly because of its initialization step that does not necessarily use all the available supervision. Though both CONSTRAINED-KMEANS and COP-KMeans treat the seeds as constraints, the fact that CONSTRAINED-KMEANS uses all the seeds to initialize clusters, as opposed to COP-KMeans which does not necessarily do that, results in the former having better performance in most cases with zero-noise. In fact, the effect of seeding seems to be so important that in some cases (Fig. 3.5), SEEDED-KMEANS performs significantly better than COP-KMeans.

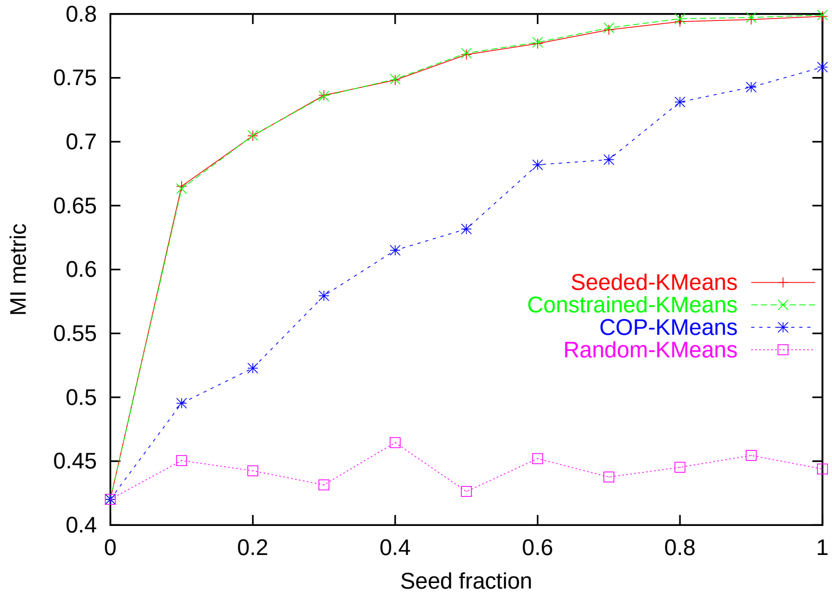

Figure 3.5: Comparison of NMI on *20-Newsgroups-100* data, noise fraction = 0

Objective function with respect to seeding: Though the NMI measure increases with an increase in seed fraction for the semi-supervised algorithms, the behavior of the objective function will depend on whether the user bias provided by the user-labeled seeds is consistent with the assumptions of KMeans. If the category structure created by the user-labeling of the data set satisfies the KMeans assumptions, then the data partition induced by seeding will be close to the optimal partition, and KMeans is known to converge to a good local optimum in this case (Fig. 3.6) (Devroye et al., 1996). On the other hand, if the user bias is inconsistent with the KMeans assumptions, then constrained seeding will result in convergence to a sub-optimal solution (Figs. 3.4, 3.8). Note that since SEEDED-KMEANS does not necessarily maintain the same assignments for the seed points in subsequent it-

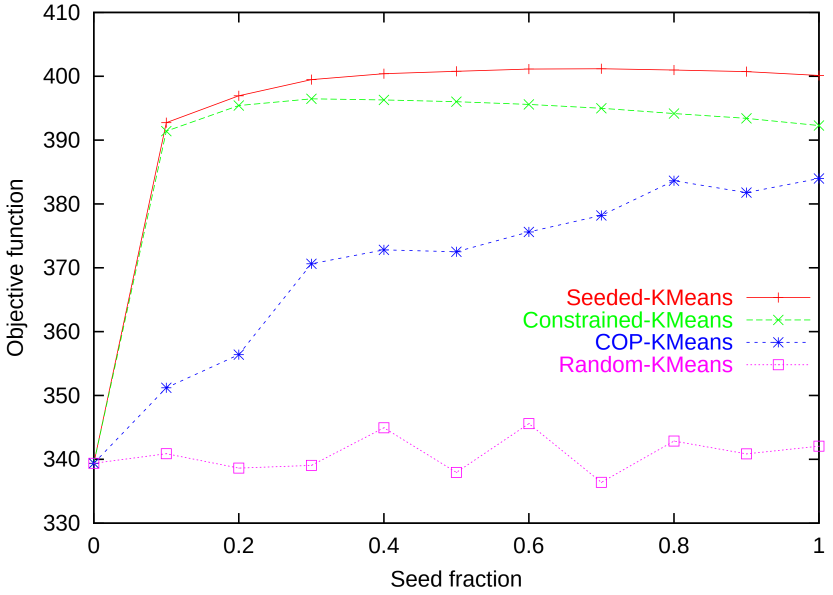

Figure 3.6: Comparison of objective function on *20-Newsgroups-100* data, noise fraction = 0

erations, its objective function does not decrease due to conflict in bias; however, since CONSTRAINED-KMEANS and COP-KMeans keep the seeds as constraints, their objective function decreases with increase in seeding. Since random KMeans never uses the seeds, its behavior is independent of this conflict.

Data Set separability: Semi-supervision gives substantial improvement over unsupervised clustering for data sets that are difficult to cluster, in the sense that the clusters are not well separated, e.g., *3-News-Similar-1000*, (Fig. 3.10). For data sets that are easily separable, e.g., *3-News-Different-1000* (Fig. 3.11) the improvement over random KMeans is marginal. If the data set is easily separable, then there are not many bad local minima and even random KMeans can easily find the cluster structure. However, for data sets with

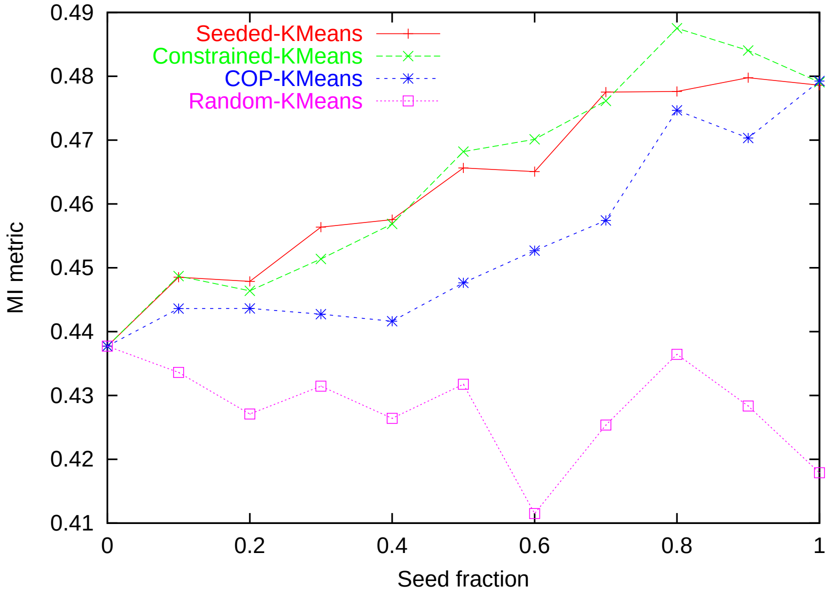

Figure 3.7: Comparison of NMI on *Yahoo! News* data, noise fraction = 0

clusters that are not well separated, seeding seems to be an important factor in helping the algorithm find a good clustering. Even with high seeding, the NMI measure for the separable data sets are in general much higher than the data sets that are not well separable, because the latter one is a harder problem to solve.

Performance with incomplete seeding: We also ran initial experiments with *incomplete* seeding, where seeds are not specified for every cluster. For these experiments, if any of the semi-supervised KMeans algorithms are run to find K clusters and we have seeds for only L clusters ($L < K$), then the remaining $K - L$ centroids are initialized by random perturbations of the global centroid, following the methodology of Dhillon et al. (Dhillon et al., 2001). From Fig. 3.12, it can be seen that the NMI metric did not decrease substan-

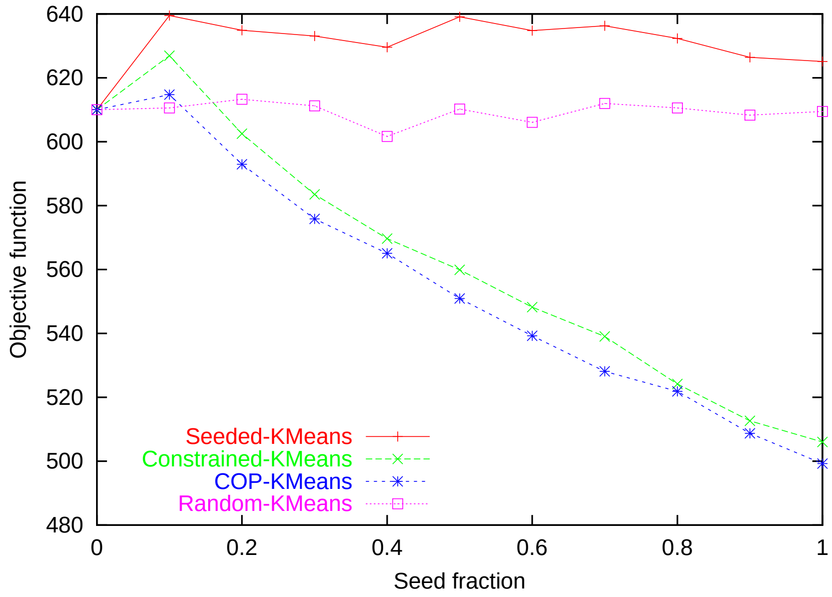

Figure 3.8: Comparison of objective function on *Yahoo! News* data, noise fraction = 0

tially with increase in the number of unseeded categories, showing that the semi-supervised clustering algorithms could extend the seed clusters and generate more clusters in order to fit the regularity of the data.

Performance with respect to noise: In many practical applications, the labeled data often has noise due to human labeling errors, inaccuracies of automated labeling processes, or other reasons. In this experiment, we study the noise robustness of all the different semi-supervised clustering algorithms, to estimate how well they would perform on real-life domains.

Fig. 3.13 shows that as noise is increased, the performance of CONSTRAINED-KMEANS and COP-KMeans starts to degrade compared to SEEDED-KMEANS. COP-

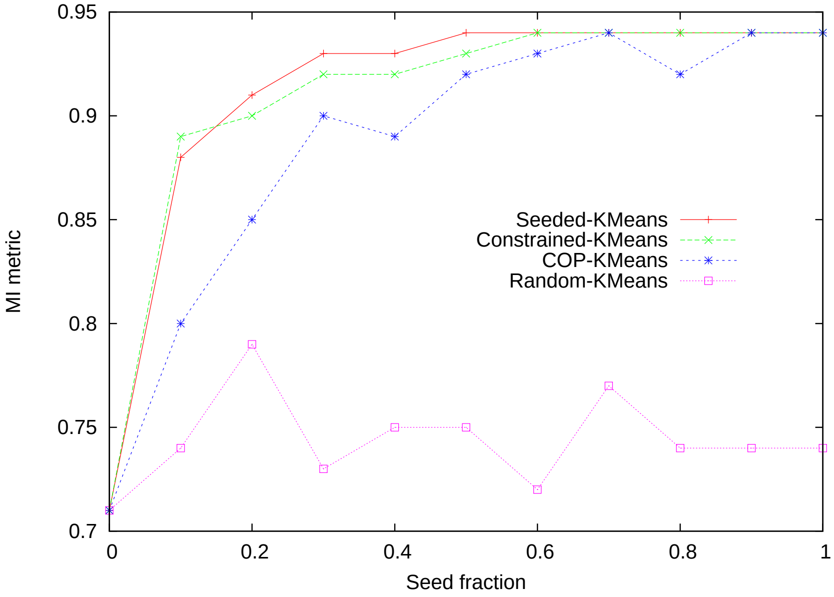

Figure 3.9: Comparison of NMI on *Iris* data, noise fraction = 0

KMeans and CONSTRAINED-KMEANS keep using the same noisy seeds in every subsequent iteration of the algorithm, whereas SEEDED-KMEANS can abandon noisy seed labels in subsequent iterations. So SEEDED-KMEANS is quite robust against noisy seeding, and can take full advantage of the seeding if it gives the algorithm a good initialization.

The statistical significance of the conclusions in this section have been tested across various data sets. For example, on the *Small-20-Newsgroup* data set, the conclusions are significant for seed fraction $>=$ 0.2 ($p < 0.001$) for the first three aspects discussed above, using two-tailed paired *t*-test. For the noise experiments, the conclusion is significant for noise fraction < 0.5 ($p < 0.001$).

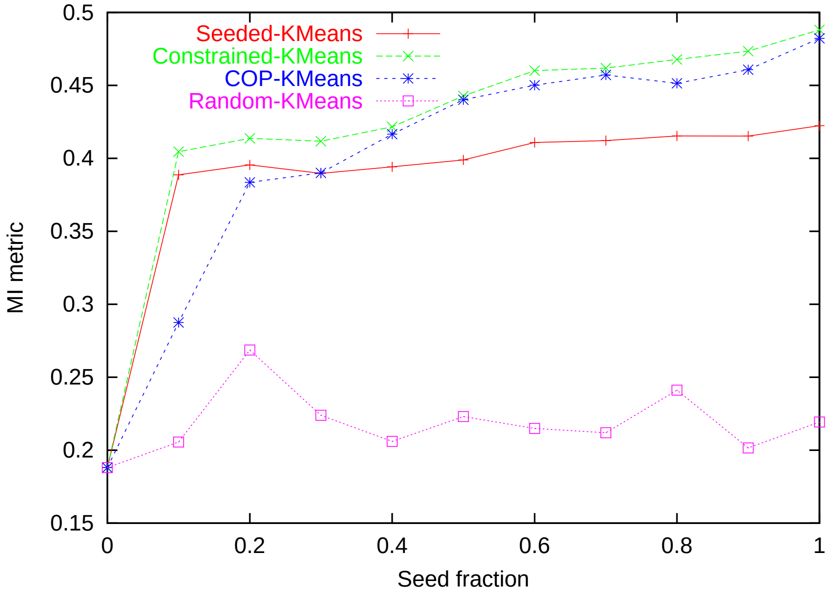

Figure 3.10: Comparison of NMI on *3-News-Similar-1000* data, noise fraction = 0

3.6 Chapter summary

In this chapter, we have shown how initial labeled data can be used to aid and bias the clustering of unlabeled data into partitions. SEEDED-KMEANS and CONSTRAINED-KMEANS are semi-supervised clustering algorithms that use labeled data to form initial clusters and constrain subsequent cluster assignment. Both methods can be viewed as instances of an EM algorithm over a mixture of unit variance Gaussians under certain conditions, where labeled data provides prior information about the conditional distributions of hidden category labels. Experimental results demonstrate the advantages of these methods over standard random seeding and COP-KMeans (Wagstaff et al., 2001), an alternative semi-supervised KMeans algorithm. In particular, seeding without constraints is a robust semi-supervised

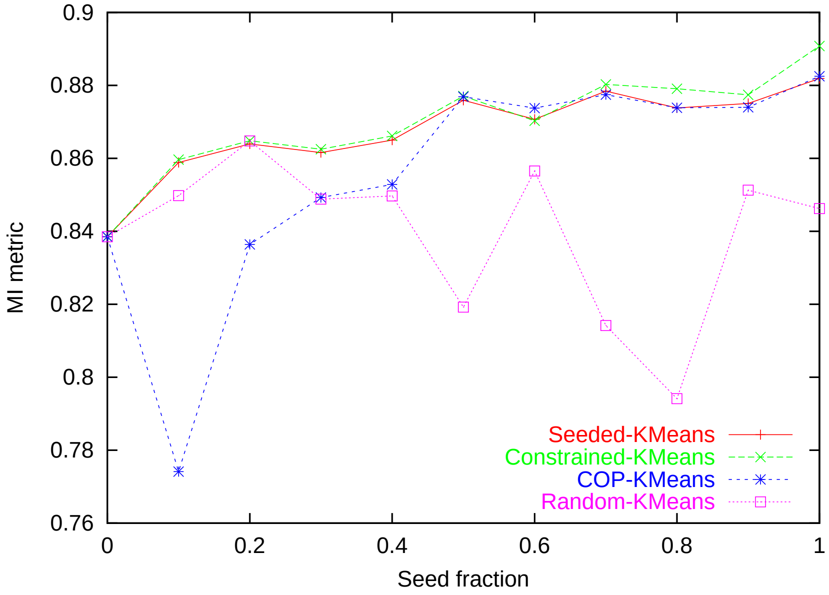

Figure 3.11: Comparison of NMI on *3-News-Different-1000* data, noise fraction = 0

method that is less sensitive to noise and imperfections in the given labeled data.

In certain applications, supervision in the form of class labels may be unavailable, while pairwise constraints on the data, specifying whether two points should be in the same cluster or in different clusters, are easily obtained. This creates the need for algorithms that can utilize such supervision – the next chapter describes one such algorithm, which can perform semi-supervised partitional clustering of data using pairwise constraints.

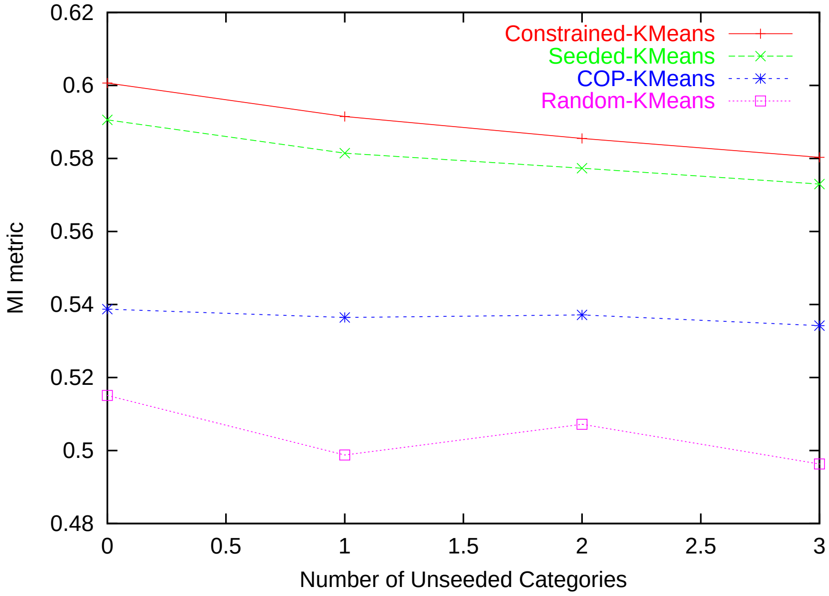

Figure 3.12: Comparison of NMI on *20-Newsgroups-1000* data, seed fraction = 0.1

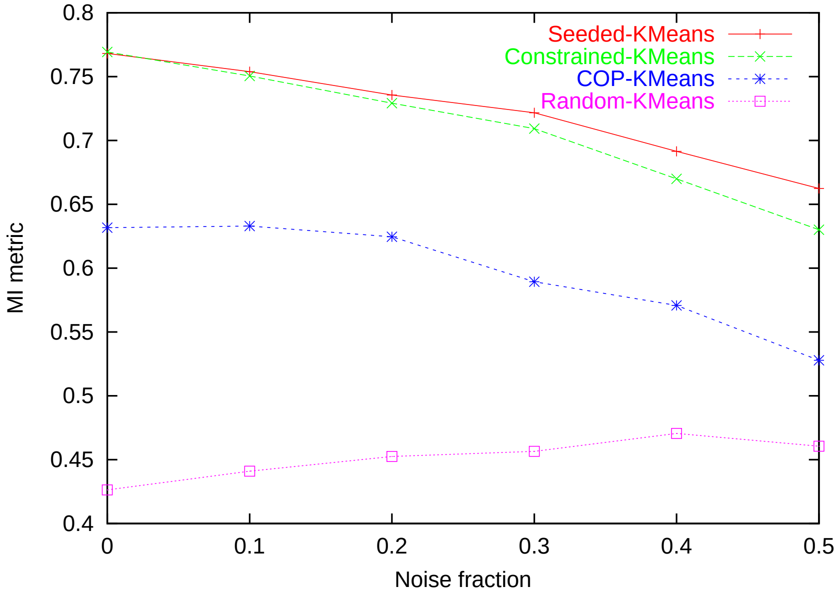

Figure 3.13: Comparison of NMI on *20-Newsgroups-100* data, seed fraction = 0.5

Chapter 4

Semi-supervised Clustering with Constraints

This chapter describes a probabilistic framework for semi-supervised clustering with pairwise constraints, based on the Hidden Markov Random Field (HMRF) model. This chapter outlines the basic HMRF model; a generalization of the model presented here, which allows integration of constraint-based and metric-based semi-supervied clustering, is discussed in Sec. 7.1.

4.1 Motivation of clustering with constraints

As mentioned in the last chapter, pairwise constraints can be a more natural form of supervision than labels in certain clustering tasks. Pairwise supervision is typically provided as *must-link* and *cannot-link* constraints on data points: a *must-link* constraint indicates that both points in the pair should be placed in the same cluster, while a *cannot-link* constraint indicates that two points in the pair should belong to different clusters. In certain applications, supervision in the form of class labels may be unavailable, while pairwise constraints are easily obtained, creating the need for methods that exploit such supervision. For example, complete class labels may be unknown in the context of clustering for speaker identification in a conversation (Bar-Hillel et al., 2003), or clustering GPS data for lane-finding (Wagstaff et al., 2001). In some domains, pairwise constraints occur naturally, e.g., the Database of Interacting Proteins (DIP) data set in biology contains information about proteins co-occurring in processes, which can be viewed as must-link constraints during clustering. Moreover, in an interactive learning setting, a user who is not a domain expert can sometimes provide feedback in the form of must-link and cannot-link constraints more easily than class labels, since providing constraints does not require the user to have significant prior knowledge about the categories in the data set.

4.2 Problem definition

Our semi-supervised clustering model with constraints considers a sample of n data points $X = \{x_i\}_{i=1}^n$, each $x_i \in \mathbb{R}^d$ being a d-dimensional vector, with x_{im} representing its m^{th} component. The model relies on a distortion measure D used to compute distance between points: $D : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Supervision is provided as two sets of pairwise constraints:

must-link constraints $C_{ML} = \{(x_i, x_j)\}$ and cannot-link constraints $C_{CL} = \{(x_i, x_j)\}$, where $(x_i, x_j) \in C_{ML}$ implies that x_i and x_j are labeled as belonging to the same cluster, while $(x_i, x_j) \in C_{CL}$ implies that x_i and x_j are labeled as belonging to different clusters. The constraints may be accompanied by associated violation costs W, where w_{ij} represents the cost of violating the constraint between points x_i and x_j if such a constraint exists, that is, either $(x_i, x_j) \in C_{ML}$ or $(x_i, x_j) \in C_{CL}$. The task is to partition the data points X into k disjoint clusters $\{X_h\}_{h=1}^{k}$ so that the total distortion between the points and the corresponding cluster representatives is (locally) minimized according to the given distortion measure D, while constraint violations are kept to a minimum.

4.3 The HMRF model

This section describes the Hidden Markov Random Field (HMRF) probabilistic model (Zhang, Brady, & Smith, 2001) for semi-supervised constrained clustering.

4.3.1 HMRF components

The HMRF model consists of the following components:

- An *observable* set $X = \{x_i\}_{i=1}^{n}$ of random variables, corresponding to the given data points X. Note that we overload notation and use X to refer to both the given set of data points and their corresponding random variables.

- An *unobservable* (hidden) set $Y = \{y_i\}_{i=1}^{n}$ of random variables, corresponding to cluster assignments of points in X. Each hidden variable y_i encodes the cluster label of the point x_i and takes values from the set of cluster indices $\{h\}_{h=1}^{k}$.

- An *unobservable* (hidden) set of generative model parameters Θ, which consists of cluster representatives $M = \{\mu_h\}_{h=1}^{k}$.
- An *observable* set of constraint variables $C = (c_{12}, c_{13}, \ldots, c_{n-1,n})$. Each c_{ij} is a tertiary variable taking on a value from the set $(-1, 0, 1)$, where $c_{ij} = 1$ indicates that $(x_i, x_j) \in C_{ML}$, $c_{ij} = -1$ indicates that $(x_i, x_j) \in C_{CL}$, and $c_{ij} = 0$ corresponds to pairs (x_i, x_j) that are not constrained.

Since constraints are fully observed and the described model does not attempt to model them generatively, the joint probability of X, Y, and Θ is conditioned on the constraints encoded by C. Fig. 4.1 shows a simple example of an HMRF. X consists of five data points with corresponding variables $(x_1, \ldots, x_5)$ that have cluster labels $Y = (y_1, \ldots, y_5)$, which may each take on values $(1, 2, 3)$ denoting the three clusters. Three pairwise constraints are provided: two must-link constraints (x_1, x_2) and (x_1, x_4), and one cannot-link constraint (x_2, x_3). Corresponding constraint variables are $c_{12} = 1$, $c_{14} = 1$, and $c_{23} = -1$; all other variables in C are set to zero. The task is to partition the five points into three clusters. Fig. 4.1 demonstrates one possible clustering configuration which does not violate any constraints. The must-linked points x_1, x_2 and x_4 belong to cluster 1; the point x_3, which is cannot-linked with x_2, is assigned to cluster 2; x_5, which is not involved in any constraints, belongs to cluster 3.

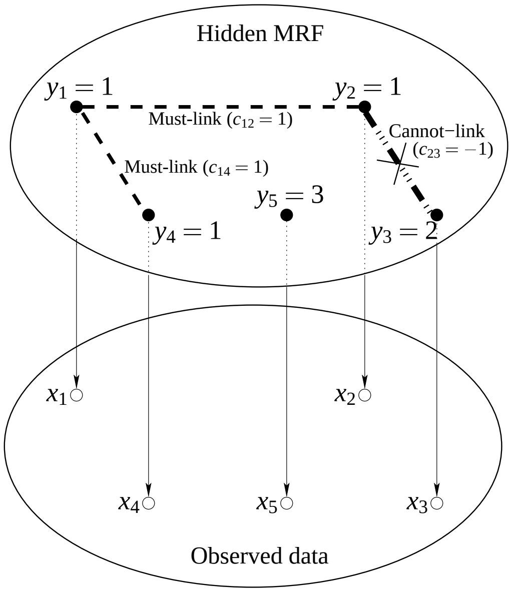

Figure 4.1: A Hidden Markov Random Field

4.3.2 Markov Random Field over labels

Each hidden random variable $y_i \in Y$, representing the cluster label of $x_i \in X$, is associated with a set of neighbors N_i. The set of neighbors is defined as all points to which x_i is must-linked or cannot-linked: $N_i = \{y_j | (x_i, x_j) \in C_{ML}$ or $(x_i, x_j) \in C_{CL}\}$. The resulting random field defined over the hidden variables Y is a Markov Random Field (MRF) (Geman & Geman, 1984), where the conditional probability distribution over the hidden variables obeys the Markov property:

$$\forall i,\ \Pr(y_i | Y - \{y_i\}, \Theta, C) = \Pr(y_i | \{y_j : y_j \in N_i\}, \Theta, C). \tag{4.1}$$

Thus the conditional probability of y_i for each x_i, given the model parameters and the set of constraints, depends only on the cluster labels of the observed variables that are must-linked or cannot-linked to x_i. Then, by the Hammersley-Clifford theorem (Hammersley & Clifford, 1971), the prior probability of a particular label configuration Y can be expressed as a Gibbs distribution (Geman & Geman, 1984), so that

$$\Pr(Y|\Theta,C) = \frac{1}{Z}\exp\big(-v(Y)\big) = \frac{1}{Z}\exp\big(-\sum_{N_i \in N} v_{N_i}(Y)\big), \tag{4.2}$$

where N is the set of all neighborhoods, Z is the normalizing term, and $v(Y)$ is the overall label configuration potential function, which can be factored into the functions $v_{N_i}(Y)$ that denote the potentials for all neighborhoods N_i in the label configuration Y. Since the potentials for all neighborhoods are based on pairwise constraints in C (and model parameters Θ), we can further factor the label configuration as:

$$\Pr(Y|C) = \frac{1}{Z}\exp(-\sum_{i,j} v(i,j)), \tag{4.3}$$

where each constraint potential function $v(i,j)$ has the following form:

$$v(i,j) = \begin{cases} w_{ij} & \text{if } c_{ij} = 1 \text{ and } y_i \neq y_j \\ w_{ij} & \text{if } c_{ij} = -1 \text{ and } y_i = y_j \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

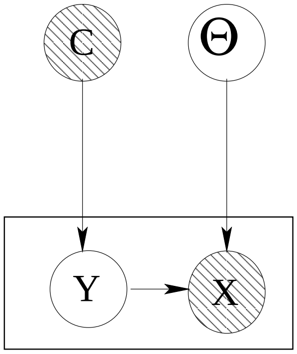

Figure 4.2: Graphical plate model of variable dependence

This constraint potential corresponds to the generalized Potts potential function (Boykov, Veksler, & Zabih, 1998; Kleinberg & Tardos, 1999). Overall, this formulation for observing the label assignment Y results in higher probabilities being assigned to configurations in which cluster assignments do not violate the provided constraints.

4.3.3 Joint probability in HMRF

The joint probability of X, Y, and Θ, given C, in the described HMRF model can be factorized as follows:

$$\Pr(X,Y,\Theta|C) = \Pr(\Theta|C)\ \Pr(Y|\Theta,C)\ \Pr(X|Y,\Theta,C) \tag{4.5}$$

The graphical plate model (Buntine, 1994) of the dependence between the random variables in the HMRF is shown in Fig. 4.2, where the clear nodes represent the hidden variables, the shaded nodes are the observed variables, the directed links show dependencies between the variables, while the lack of an edge between two variables implies conditional independence. The prior over the parameters Θ is independent of the constraints C, i.e., $P(\Theta|C) = P(\Theta)$. The probability of observing the label configuration Y depends on the constraints C but is independent of the current generative model parameters Θ, so that $P(Y|\Theta,C) = P(Y|C)$. Observed data points corresponding to variables X are generated using the model parameters Θ based on cluster labels Y and are independent of the constraints C, so that $P(X|Y,\Theta,C) = P(X|Y,\Theta)$. The variables X are assumed to be mutually independent: each x_i is generated individually from a conditional probability distribution $\Pr(x|y,\Theta)$. Then, the conditional probability $\Pr(X|Y,\Theta,C)$ can be written as:

$$\Pr(X|Y,\Theta,C) = \Pr(X|Y,\Theta) = \prod_{i=1}^{n} p(x_i|y_i,\Theta), \tag{4.6}$$

where $p(\cdot|y_i,\Theta)$ is the probability density function for the y_i^{th} cluster, from which x_i is generated. This probability density is related to the clustering distortion measure D, as described in Sec. 4.3.4.

From Eqns. (4.3), (4.5), and (4.6), and using the independence assumptions, it follows that maximizing the joint probability on the HMRF is equivalent to maximizing:

$$\Pr(X,Y,\Theta|C) = \Pr(\Theta)\left(\frac{1}{Z}\exp\big(-\sum_{c_{ij}\in C} v(i,j)\big)\right)\left(\prod_{i=1}^{n} p(x_i|y_i,\Theta)\right) \tag{4.7}$$

The joint probability in Eqn. (4.7) has 3 factors. The first factor describes a prior probability distribution over the model parameters. The second factor is the conditional probability of observing a particular label configuration given the provided constraints, effectively assigning a higher probability to configurations where the cluster assignments do not violate the constraints. Finally, the third factor is the conditional probability of generating the observed data points given the labels and the parameters: if *maximum likelihood* (ML) estimation was performed on the HMRF, the goal would have been to maximize this term in isolation.

Overall, maximizing the joint HMRF probability in Eqn. (4.7) is equivalent to jointly maximizing the likelihood of generating data points from the model and the probability of label assignments that respect the constraints.

4.3.4 Semi-supervised clustering objective function on HMRF

Eqn. (4.7) suggests a general framework for incorporating constraints into clustering. A particular choice of the conditional probability $p(\cdot|y,\Theta)$ is directly connected to the choice of the distortion measure appropriate for the clustering task.

When considering the conditional probability $p(\cdot|y,\Theta)$ – the probability of generating a data point from the y^{th} cluster – we restrict our attention to probability densities from the exponential family, where the expectation parameter corresponding to the h^{th} cluster is μ_h, the mean of the points of that cluster. Using this assumption and the bijection between regular exponential distributions and regular Bregman divergence (Banerjee et al., 2004), the conditional density for observed data can be represented as:

$$p(x_i|y_i,\Theta) = \frac{1}{Z_\Theta}\exp\big(-D(x_i,\mu_h)\big), \tag{4.8}$$

where $D(x_i, \mu_h)$ is the Bregman divergence between x_i and μ_h, corresponding to the exponential density p, and Z_Θ is the normalizer. Different clustering models fall into this exponential form:

- If x_i and μ_h are vectors in Euclidean space, and D is the square of the L_2 distance $\big(D(x_i, \mu_h) = \|x_i - \mu_h\|^2\big)$, then the cluster conditional probability is a Gaussian with unit covariance (Kearns et al., 1997);

- If x_i and μ_h are probability distributions and D is the KL-divergence $\big(D(x_i, \mu_h) = \sum_{m=1}^{d} x_{im} \log \frac{x_{im}}{\mu_{hm}}\big)$, then the cluster conditional probability is a multinomial distribution (Dhillon & Guan, 2003).

The relation in Eqn. (4.8) holds even if D is not a Bregman divergence but a directional distance measure like cosine distance. For example, if x_i and μ_h are vectors of unit length and D is one minus the dot-product of the vectors $\big(D(x_i, \mu_h) = 1 - \frac{\sum_{m=1}^{d} x_{im}\mu_{hm}}{\|x_i\|\|\mu_h\|}\big)$, then the cluster conditional probability is a von-Mises Fisher (vMF) distribution with unit concentration parameter (Banerjee et al., 2003), which is essentially the spherical analog of a Gaussian.

Putting Eqn. (4.8) into Eqn. (4.7) and taking logarithms gives the following cluster objective function, minimizing which is equivalent to maximizing the joint probability over the HMRF in Eqn. (4.7):

$$\mathcal{J}_{\text{hmrf-kmeans}} = \sum_{x_i \in X} D(x_i, \mu_{y_i}) + \sum_{c_{ij} \in C} v(i, j) - \log \Pr(\Theta) + \log Z + \log Z_\Theta \qquad (4.9)$$

Thus, the task is to minimize $\mathcal{J}_{\text{hmrf-kmeans}}$ over the hidden variables Y and Θ (note that given Y, the means $M = \{\mu_h\}_{h=1}^{k}$ are uniquely determined).

4.4 The HMRF-KMeans algorithm

Since the cluster assignments and the generative model parameters are unknown in a clustering setting, minimizing Eqn. (4.9) is an "incomplete-data problem". A popular solution technique for such problems the is *Expectation Maximization* (EM) algorithm (Dempster et al., 1977). The KMeans algorithm (MacQueen, 1967) is known to be equivalent to the EM algorithm with hard clustering assignments, under certain assumptions (Kearns et al., 1997; Basu et al., 2002; Banerjee et al., 2004). This section describes a KMeans-type hard partitional clustering algorithm, HMRF-KMEANS, that finds a local minimum of the semi-supervised clustering objective function $\mathcal{J}_{\text{hmrf-kmeans}}$ in Eqn. (4.9).

4.4.1 Approximations

Before describing the details of the clustering algorithm, it is important to consider the normalizer components: the MRF normalizer $\log Z$ and the distortion function normalizer $\log Z_{\Theta}$ in Eqn. (4.9). Estimation of the MRF normalizer cannot be performed in closed form, and approximate inference methods must be employed for computing it (Wainwright & Jordan, 2003). Estimation of the distortion normalizer $\log Z_{\Theta}$ depends on the distortion measure D used by the model. This chapter considers three distortion measures: squared

Euclidean distance, cosine distance, and Kullback-Leibler (KL) divergence. For Euclidean distance, Z_Θ can be estimated in closed form, and this estimation is performed while minimizing the clustering objective function $\mathcal{J}_{\text{hmrf-kmeans}}$ in Eqn. (4.9). For the other distortion measures, estimating the distortion normalizer Z_Θ cannot be performed in closed form, and approximate inference must be again used.

Since approximation methods can be very expensive computationally, two simplifying assumptions can be made: the MRF normalizer may be considered to be constant in the clustering process, and the distortion normalizer may be assumed constant for all distortion measures that do not provide its closed-form estimate. With these assumptions, the objective function $\mathcal{J}_{\text{hmrf-kmeans}}$ in Eqn. (4.9) no longer exactly corresponds to a joint probability on a HMRF. However, minimizing this simplified objective has been shown to work well empirically (Bilenko, Basu, & Mooney, 2004; Basu, Bilenko, & Mooney, 2004). However, if in some application it is important to preserve the semantics of the underlying joint probability model, then the normalizers Z and Z_Θ must be estimated by approximation methods.

The prior term $\log \Pr(\Theta)$, which was present in Eqn. (4.9) and the subsequent equations, can be expressed as follows:

$$\log \Pr(\Theta) = \log\big(\Pr(M)\big).$$

The prior $\Pr(M)$ over the cluster centroids is assumed to be uniform, and so this term can be dropped from $\mathcal{J}_{\text{hmrf-kmeans}}$. With these approximations, the semi-supervised clustering objective function can be expressed as:

$$\mathcal{J}_{\text{hmrf-kmeans}} \;=\; \sum_{x_i \in X} D(x_i, \mu_{y_i}) + \sum_{\substack{(x_i,x_j)\in C_{ML} \\ s.t.\ y_i \neq y_j}} w_{ij} + \sum_{\substack{(x_i,x_j)\in C_{CL} \\ s.t.\ y_i = y_j}} w_{ij}. \qquad (4.10)$$

4.4.2 EM framework

$\mathcal{J}_{\text{hmrf-kmeans}}$ can be (locally) minimized by a KMeans-type iterative algorithm that we call HMRF-KMEANS. The outline of the algorithm is presented in Fig. 4.3. The basic idea of HMRF-KMEANS is as follows: the constraints are used to get good initialization of the clustering. Then in the E-step, given the current cluster representatives, every data point is re-assigned to the cluster which minimizes its contribution to $\mathcal{J}_{\text{hmrf-kmeans}}$. In the M-step, the cluster representatives M are re-estimated from the cluster assignments to minimize $\mathcal{J}_{\text{hmrf-kmeans}}$ for the current assignment.

Effectively, the E-step minimizes $\mathcal{J}_{\text{hmrf-kmeans}}$ over cluster assignments Y, and the M-step minimizes $\mathcal{J}_{\text{hmrf-kmeans}}$ over cluster representatives M. The E-step and the M-step are repeated till a specified convergence criterion is reached. The specific details of the E-step and M-step are discussed in the following sections.

4.4.3 Initialization

Good initial centroids are essential for the success of partitional clustering algorithms such as KMeans. For HMRF-KMEANS, a two stage initialization process is used to get good

```
Algorithm: HMRF-KMeans
Input: Set of data points $X = \{x_i\}_{i=1}^n$, number of clusters $k$, set of constraints
    $C$, constraint violation costs $W$, distortion measure $D$.
Output: Disjoint $k$-partitioning $\{X_h\}_{h=1}^k$ of $X$ such that objective function $\mathcal{J}_{\text{hmrf-kmeans}}$
    in Eqn. (4.10) is locally minimized.
Method:
1. Initialize the $k$ clusters centroids $\{\mu_h^{(0)}\}_{h=1}^k$ using $C$, set t $\leftarrow$ 0.
2. Repeat until convergence
2a.    E-step: Given centroids $M^{(t)} = \{\mu_h^{(t)}\}_{h=1}^k$, re-assign cluster labels
       $Y^{(t+1)} = \{y_i^{(t+1)}\}_{1=1}^n$ on $X$ to minimize $\mathcal{J}_{\text{hmrf-kmeans}}$.
2b.    M-step: Given cluster labels $Y^{(t+1)}$, re-calculate centroids $M^{(t+1)}$
       to minimize $\mathcal{J}_{\text{hmrf-kmeans}}$.
2c.    $t \leftarrow (t+1)$
```

Figure 4.3: HMRF-KMeans algorithm

centroids from both the constraints and the unlabeled data.

Neighborhood inference: At first, the transitive closure of the must-link constraints is taken to get connected components consisting of points connected by must-links. Let there be λ connected components, which are used to create λ neighborhoods. These correspond to the must-link neighborhoods in the MRF over the hidden cluster variables.

Cluster selection: The λ neighborhood sets produced in the first stage are used to initialize the HMRF-MEANS algorithm. If $\lambda = k$, λ cluster centers are initialized with the centroids of all the λ neighborhood sets. If $\lambda < k$, λ clusters are initialized from the neighborhoods, and the remaining $k - \lambda$ clusters are initialized with points obtained by random perturbations of the global centroid of X, following the methodology of Dhillon et al. (Dhillon et al., 2001). If $\lambda > k$, a weighted variant of farthest-first traversal (Hochbaum & Shmoys, 1985) is applied to the centroids of the λ neighborhoods, where the weight of each centroid is proportional to the size of the corresponding neighborhood. Weighted farthest-first traversal selects neighborhoods that are relatively far apart as well as large in size, and the chosen neighborhoods are set as the k initial cluster centroids for HMRF-KMEANS.

Overall, this two-stage initialization procedure is able to take into account both unlabeled data and constraints to obtain cluster representatives that provide a good initial partitioning of the data set.

4.4.4 E-step

In the E-step, assignments of data points to clusters are updated using the current estimates of the cluster representatives. In the general unsupervised KMeans algorithm, there is no interaction between the cluster labels, and the E-step is a simple assignment of every point to the cluster representative that is nearest to it according to the clustering distortion measure. In contrast, the HMRF model incorporates interaction between the cluster labels defined by the random field over the hidden variables. As a result, computing the assignment of data points to cluster representatives to find the global minimum of the objective function, given the cluster centroids, is computationally intractable in any non-trivial HMRF model (Segal, Wang, & Koller, 2003a).

There exist several techniques for computing cluster assignments that approximate the optimal solution in this framework. In this section we follow the iterated conditional modes (ICM) approach (Besag, 1986; Zhang et al., 2001), which is a greedy strategy to sequentially update the cluster assignment of each point while keeping the assignments for the other points fixed. Global methods of collective inference in the E-step include loopy belief propagation (Pearl, 1988; Segal et al., 2003a) and linear programming relaxation (Kleinberg & Tardos, 1999), which are described in Appendix A.1 and A.2 respectively. As will be shown by experiments in Sec. 4.5.4, the inexpensive greedy ICM algorithm gives a clustering accuracy that is comparable to the expensive global approximation techniques and it is computationally more efficient.

ICM performs sequential cluster assignment for all the points in random order. Each point x_i is assigned to the cluster representative μ_h that minimizes the point's contribution to the objective function $\mathcal{J}_{\text{hmrf-kmeans}}(x_i, \mu_h)$:

$$\mathcal{J}_{\text{hmrf-kmeans}}(x_i, \mu_h) = D(x_i, \mu_h) + \sum_{\substack{(x_i,x_j) \in C^i_{ML} \\ s.t.\ y_i \neq y_j}} w_{ij} + \sum_{\substack{(x_i,x_j) \in C^i_{CL} \\ s.t.\ y_i = y_j}} w_{ij}, \tag{4.11}$$

where C^i_{ML} and C^i_{CL} are the subsets of C_{ML} and C_{CL} respectively in which x_i appears in the constraints.

The optimal assignment for every point minimizes the distortion between the point and its cluster representative (first term of $\mathcal{J}_{\text{hmrf-kmeans}}$) along with incurring a minimal penalty for constraint violations caused by this assignment (second term of $\mathcal{J}_{\text{hmrf-kmeans}}$). After all points are assigned, they are randomly re-ordered, and the assignment process is repeated. This process proceeds until no point changes its cluster assignment between two successive iterations.

Overall, the assignment of points to clusters incorporates pairwise supervision by discouraging constraint violations while minimizing the distance between the points and their corresponding centroids, thereby getting a desirable partitioning of the data.

4.4.5 M-step

In the M-step, the cluster centroids M are re-estimated from points currently assigned to them, to decrease the objective function $\mathcal{J}_{\text{hmrf-kmeans}}$ in Eqn. (4.10). For Bregman divergences and cosine distance, the cluster representative calculated in the M-step of the EM algorithm is equivalent to the expectation value over the points in that cluster, which is equal to their arithmetic mean (Banerjee et al., 2003, 2004). Additionally, it has been experimentally demonstrated that while clustering with distribution-based measures, e.g., KL divergence (d_{KL}), smoothing cluster representatives by a prior using a deterministic annealing schedule leads to considerable improvements (Dhillon & Guan, 2003). With smoothing controlled by a parameter α, each cluster representative μ_h is estimated as follows when d_{KL} is the distortion measure:

$$\mu_h^{(KL)} = \frac{1}{1+\alpha}\left(\frac{\sum_{x_i \in X_h} x_i}{|X_h|} + \alpha\frac{1}{n}\right) \tag{4.12}$$

For directional measures like cosine distance (d_{cos}), each cluster representative is the arithmetic mean projected onto unit sphere (Banerjee et al., 2003). Centroids are estimated as follows when d_{cos} is the distortion measure:

$$\frac{\mu_h^{(cos)}}{\|\mu_h^{(cos)}\|} = \frac{\sum_{x_i \in X_h} x_i}{\|\sum_{x_i \in X_h} x_i\|} \tag{4.13}$$

4.4.6 Convergence of HMRF-KMeans

Theorem: *The* HMRF-KMeans *algorithm converges to a local minima of* $\mathcal{J}_{hmrf\text{-}kmeans}$.

Proof: The HMRF-KMeans algorithm alternates between updating the assignment of points to clusters and updating the cluster centroids. Since all updates ensure a decrease in the objective function, each iteration of HRMF-KMEANS monotonically decreases the objective function (or it remains the same). Let us inspect each step in the update to ensure that this is indeed the case.

For analyzing the cluster assignment step, let us consider Eqn. (4.10). Each point x_i moves to a new cluster h only if the following component, contributed by the point x_i, is decreased with the move:

$$D(x_i, \mu_h) + \sum_{\substack{(x_i,x_j)\in C_{ML}^i \\ s.t.\ y_i \neq y_j}} w_{ij} + \sum_{\substack{(x_i,x_j)\in C_{CL}^i \\ s.t.\ y_i = y_j}} w_{ij}.$$

Given a set of centroids, the new cluster assignment of points will decrease $\mathcal{J}_{\text{hmrf-kmeans}}$ or keep it unchanged.

For analyzing the centroid re-estimation step, let us consider an equivalent form of Eqn. (4.10):

$$\mathcal{J}_{\text{hmrf-kmeans}} = \sum_{h=1}^{k} \sum_{x_i \in X_h} D(x_i, \mu_h) + \sum_{\substack{(x_i,x_j)\in C_{ML}^i \\ s.t.\ y_i \neq y_j}} w_{ij} + \sum_{\substack{(x_i,x_j)\in C_{CL}^i \\ s.t.\ y_i = y_j}} w_{ij}. \tag{4.14}$$

Each cluster centroid μ_h is re-estimated by taking the mean of the points in the partition X_h, which minimizes the component $\sum_{x_i \in X_h} D(x_i, \mu_h)$ of $\mathcal{J}_{\text{hmrf-kmeans}}$ in Eqn. (4.14) contributed by the partition X_h for any Bregman divergence D (Banerjee et al., 2004). The constraint potential and the prior term in the objective function do not take a part in centroid re-estimation, because they are not functions of the centroid. So, given the cluster assignments,

$\mathcal{J}_{\text{hmrf-kmeans}}$ will decrease or remain the same in this step.

Hence the objective function decreases (or remains the same) after every cluster assignment and centroid re-estimation step. Now, note that the objective function is bounded below by a constant. Being the negative log-likelihood of a probabilistic model with the normalizer terms, $\mathcal{J}_{\text{hmrf-kmeans}}$ is bounded below by zero. Even without the normalizers, the objective function is bounded below by zero, since the distortion and potential terms are non-negative. Since $\mathcal{J}_{\text{hmrf-kmeans}}$ is bounded below, and HMRF-KMEANS results in a decreasing sequence of objective function values, the value sequence must have an accumulation point. The accumulation point in this case will be a fixed point of $\mathcal{J}_{\text{hmrf-kmeans}}$ since neither updating the assignments or the centroids can further decrease the value of the objective function. As a result, the HMRF-KMEANS algorithm will converge to a fixed point (local minimum) of the objective. In practice, convergence can be determined if subsequent iterations of HMRF-KMEANS result in insignificant changes in $\mathcal{J}_{\text{hmrf-kmeans}}$. ■

4.5 Experiments

This section describes the experiments we performed to demonstrate the effectiveness of HMRF-KMEANS.

4.5.1 Data sets

Experiments were conducted on 3 data sets from the UCI repository (Blake & Merz, 1998): *Iris*, and randomly sampled subsets from the *Digits* and *Letters* handwritten character recognition data sets. *Iris* is the same data set that was described in Sec. 3.5.1. For *Digits* and *Letters*, we chose two sets of 3 classes each: $\{$**I, J, L**$\}$ from *Letters* and $\{$**3, 8, 9**$\}$ from *Digits*, sampling 10% of the data points from the original data sets randomly. These classes were chosen since they represent difficult visual discrimination problems. *Digits* has 317 data points in 16 dimensions, and *Letters* has 227 points in 16 dimensions.

When clustering sparse high-dimensional data, e.g., text documents represented using the vector space model, it is particularly difficult to cluster small data sets. This is due to the fact that clustering algorithms can easily get stuck in local optima on such data sets, which leads to poor clustering quality. In previous studies with SP-KMeans algorithm applied to document collections whose size is small compared to the dimensionality of the word space, it has been observed that there is little relocation of documents between clusters for most initializations, which leads to poor clustering quality after convergence of the algorithm (Dhillon & Guan, 2003).

This scenario is likely in many realistic applications. For example, when clustering the search results in a web-search engine like Vivísimo,[1] typically the number of webpages that are being clustered is in the order of hundreds. However the dimensionality of the

[1] http://www.vivisimo.com

feature space, corresponding to the number of unique words in all the webpages, is in the order of thousands. Moreover, each webpage is sparse, since it contains only a small number of all the possible words. Supervision in the form of pairwise constraints (e.g., must-link constraints derived from co-occurrence statistics in weblogs) can be beneficial in such cases and may significantly improve clustering quality.

To demonstrate the effectiveness of our semi-supervised clustering framework, we consider 3 data subsets *3-News-Different-100*, *3-News-Related-100* and *3-News-Similar-100* derived from the *20-Newsgroups* data set. The only difference of the 3-newsgroup data subsets from the ones described in Sec. 3.5.1 is that these subsets were derived from the reduced data set *Small-20-Newsgroups*, while the data subsets explained in Sec. 3.5.1 were derived from the original *20-Newsgroups* data set.

These 3 data subsets we use in these experiments have the characteristics of being sparse, high-dimensional, as well as having a small number of points compared to the dimensionality of the space. The vector-space model of *3-News-Similar-100* has 300 points in 1864 dimensions, *3-News-Related-100* has 300 points in 3225 dimensions, and *3-News-Different-100* had 300 points in 3251 dimensions. The clusters in *3-News-Different-100* are more well-separated than those in *3-News-Similar-100* and *3-News-Related-100*.

4.5.2 Methodology

We generated learning curves using 20 runs of 2-fold cross-validation for each data set for studying the effect of constraints in clustering: we selected 50% of the data set to be set aside as the test set at any particular fold, so that on small data sets the improvements are statistically significant. The different points along the learning curve correspond to constraints that are given as input to the semi-supervised clustering algorithm. These constraints are obtained from the training set corresponding to the remaining 50% of the data by randomly selecting pairs of points from the training set, and creating must-link or cannot-link constraints depending on whether the underlying classes of the two points are same or different. Unit constraint costs W were used for all constraints, original and inferred, since the data sets did not provide individual weights for the constraints. The clustering results were evaluated using the NMI measure, which was described in Sec. 2.4. The clustering algorithm was run on the whole data set, but NMI was calculated only on the test set. The learning curve results were averaged over the 20 runs.

In our experiments, we compared the proposed HMRF-KMEANS algorithm with its ablations. In these ablation studies, each component of HMRF-KMEANS was knocked-off to study the impact of that component of the algorithm. The following variants were compared:

- HMRF-KMEANS-I-C is the complete HMRF-KMEANS algorithm that includes use of supervised data in initialization (I), as described in Sec. 4.4.3, and incorporates constraints in cluster assignments (C) as described in Sec. 4.4.4;

- HMRF-KMEANS-I is an ablation of HMRF-KMEANS that uses pairwise supervision for initialization only, but does not perform constrained assignment;

- KMEANS is the unsupervised K-Means algorithm.

4.5.3 Results and discussion

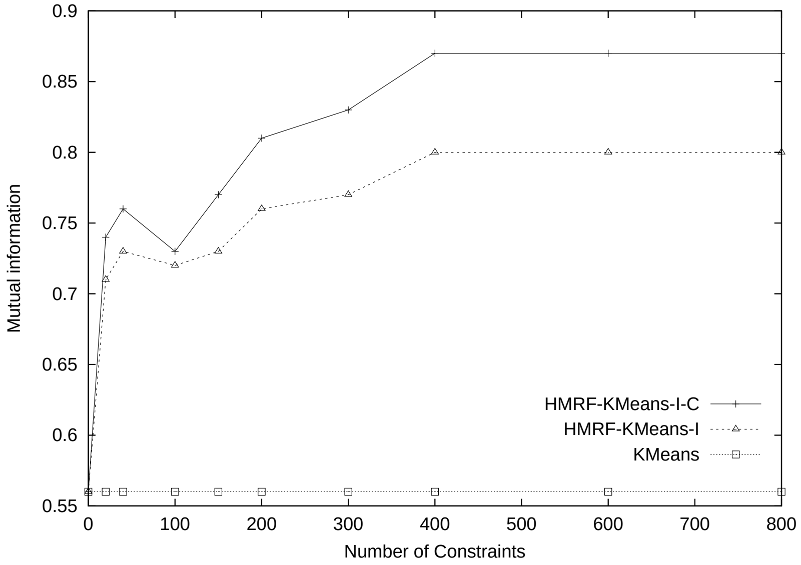

Figure 4.4: Clustering results for d_{euc} on *Iris* data set

Figs. 4.4-4.6 show the results of the ablation experiments for squared Euclidean distance d_{euc}, Figs. 4.7-4.9 demonstrate the results for experiments where cosine similarity d_{cos} was used as the distortion measure, while Figs. 4.10-4.12 show the results with KL-divergence d_{KL}.

As the results demonstrate, the full HMRF-KMEANS algorithm outperforms the ablated versions of HMRF-KMEANS for d_{euc}, d_{cos} as well as d_{KL}. On the low-dimensional data sets, the HMRF-KMEANS-I-C outperforms individual seeding (HMRF-KMEANS-I) and unsupervised clustering (KMEANS). Superiority of semi-supervised over unsupervised

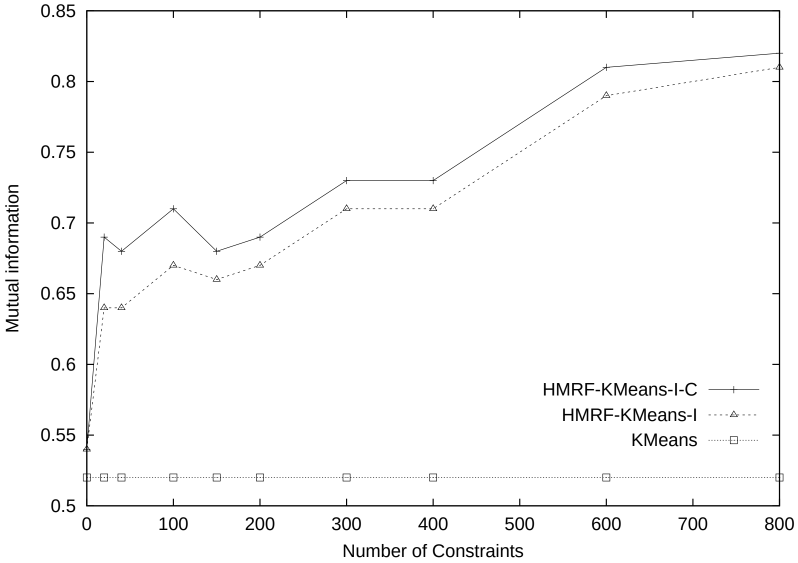

Figure 4.5: Clustering results for d_{euc} on *Digits-389* data set

clustering illustrates that providing pairwise constraints is beneficial to clustering quality. For the high-dimensional data, the relative clustering performances of HMRF-KMEANS-I-C and HMRF-KMEANS-I indicate that using supervision for initializing cluster representatives is highly beneficial, while the constraint-sensitive cluster assignment step does not lead to significant additional improvements for d_{cos}. For d_{KL}, HMRF-KMEANS-I-C outperforms HMRF-KMEANS-I on *3-News-Different-100* (Fig. 4.10) and *3-News-Similar-100* (Fig. 4.12) which indicates that incorporating constraints in the cluster assignment process is useful for these data sets. This result is reversed for *3-News-Related-100* (Fig. 4.11), implying that in some cases using constraints in the E-step may be unnecessary, which agrees with previous results on other domains (Sec. 3.5). However, incorporating supervised data in both initialization and cluster assignment always leads to substantial improvement over

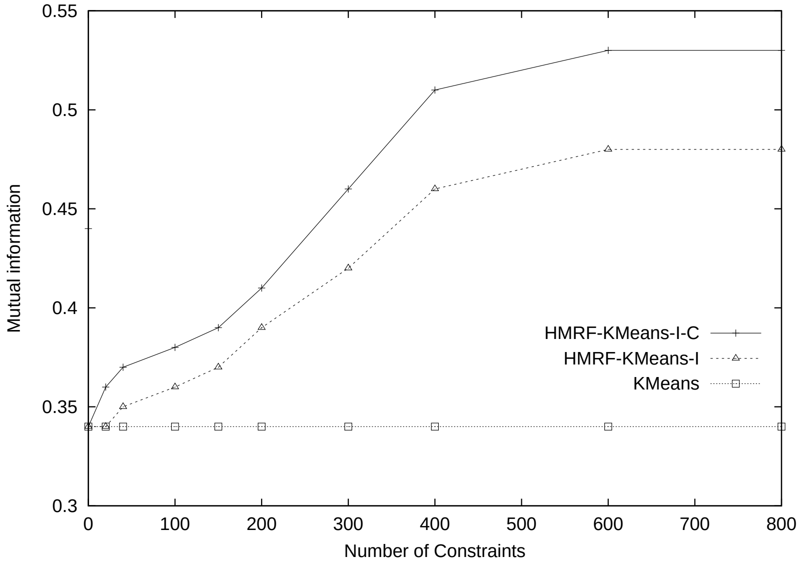

Figure 4.6: Clustering results for d_{euc} on *Letters-IJL* data set

unsupervised clustering. The improvements of the full HMRF-KMEANS over KMEANS are statistically significant on all parts of the learning curve (except for 0 constraints) for a two-tailed paired *t*-test ($p < 0.005$).

In realistic application domains, supervision in the form of constraints would be in most cases provided by human experts, in which case it is important that any semi-supervised clustering algorithm performs well with a small number of constraints. HMRF-KMEANS-I-C starts performing well early on in the learning curve, and is therefore a very appropriate algorithm to use in actual semi-supervised data clustering systems.

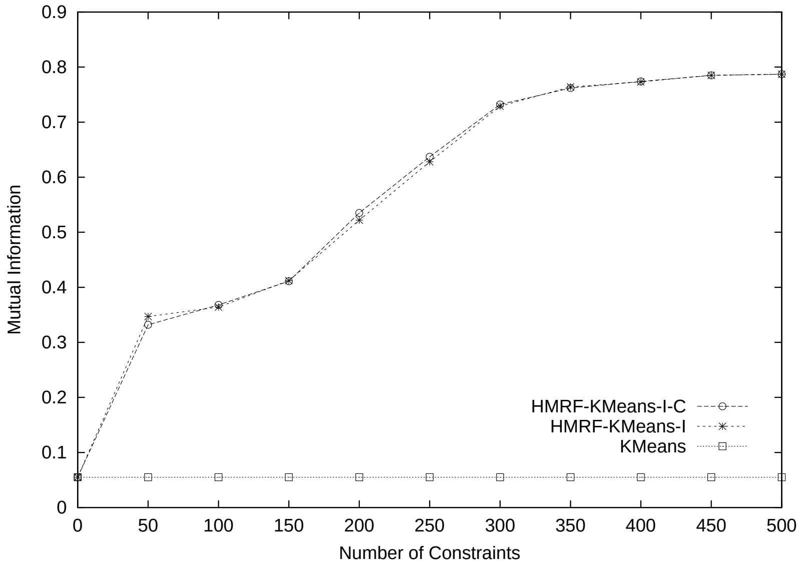

Figure 4.7: Clustering results for d_{cos} on *3-News-Different-100* data set

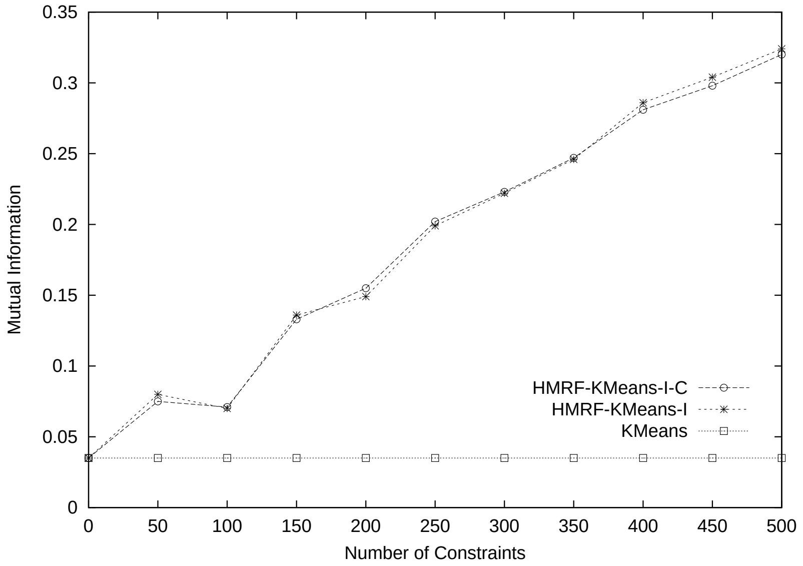

Figure 4.8: Clustering results for d_{cos} on *3-News-Related-100* data set

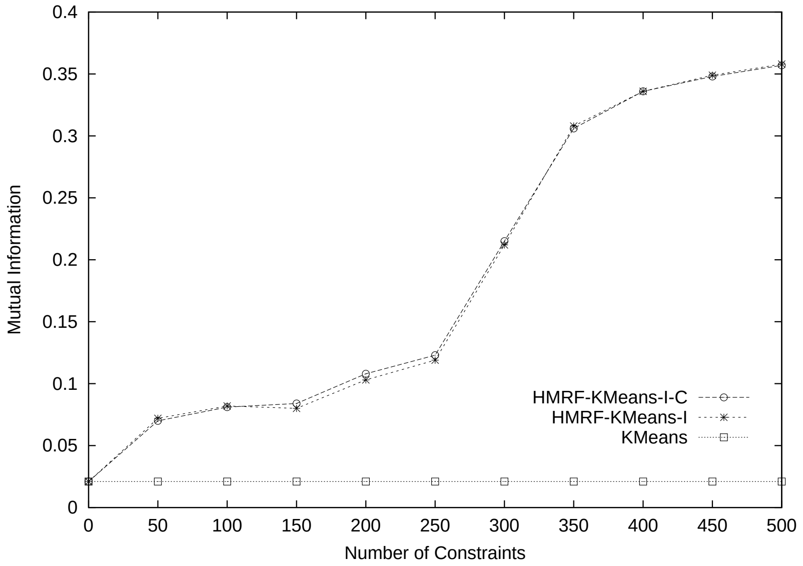

Figure 4.9: Clustering results for d_{cos} on *3-News-Similar-100* data set

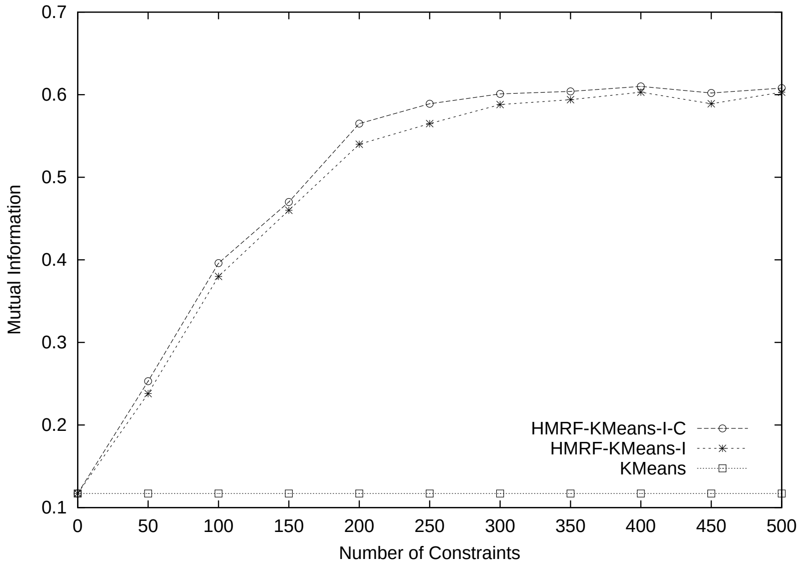

Figure 4.10: Clustering results for d_{KL} on *3-News-Different-100* data set

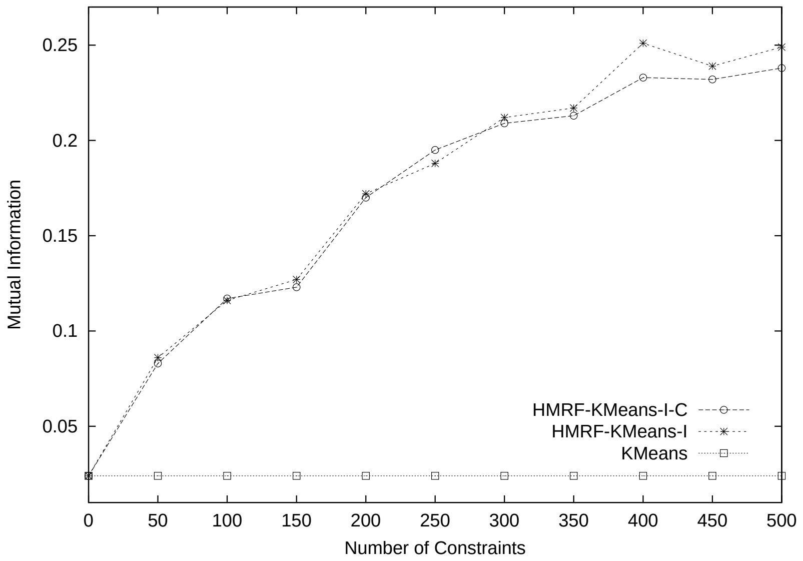

Figure 4.11: Clustering results for d_{KL} on *3-News-Related-100* data set

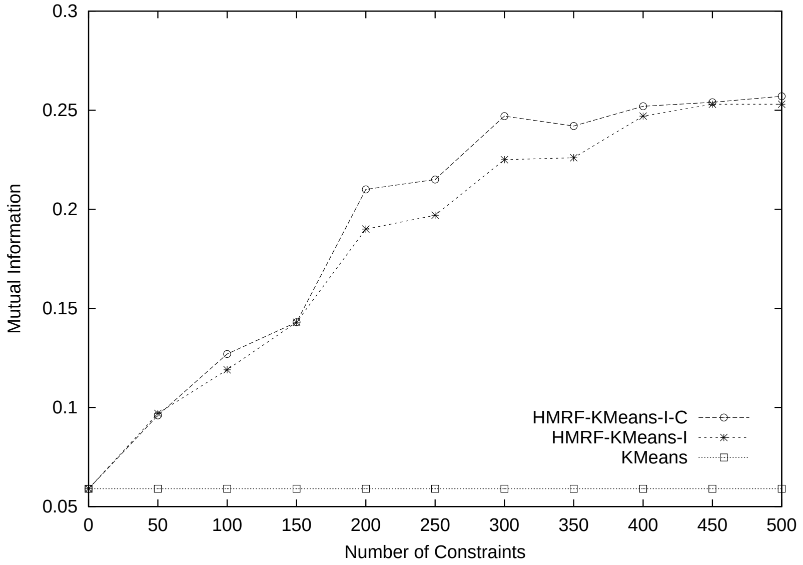

Figure 4.12: Clustering results for d_{KL} on *3-News-Similar-100* data set

4.5.4 Comparison of inference techniques

We empirically compared the greedy ICM inference technique with the two global inference techniques (loopy belief propagation and linear programming relaxation) for collective assignment of instances to clusters, the details of which are described in Appendix A. Fig. 4.13 is the learning curve for the *Iris* data set. As the graph demonstrates, global inference methods such as loopy belief propagation (BP) and linear programming (LP) relaxation outperform the greedy approaches when a limited number of pairwise constraints is provided. However, as the number of provided constraints increases, returns from these computationally expensive methods diminish; after a particular number of constraints, ICM performs no worse than the global approximate inference methods. A note on computational requirements: in our experiments, we noticed that ICM was about 10-15 times faster than the BP and LP methods for most data sets.

4.6 Chapter summary

In this chapter, we have shown how constraints can be used to improve the performance of clustering. We have a probabilistic formulation based on Hidden Markov Random Fields (HMRFs) that leads to a semi-supervised clustering objective function derived from the joint probability of observed data points, their cluster assignments, and generative model parameters. We propose an EM-style clustering algorithm, HMRF-KMEANS, that finds a local minimum of this objective function. HMRF-KMEANS can be used to perform

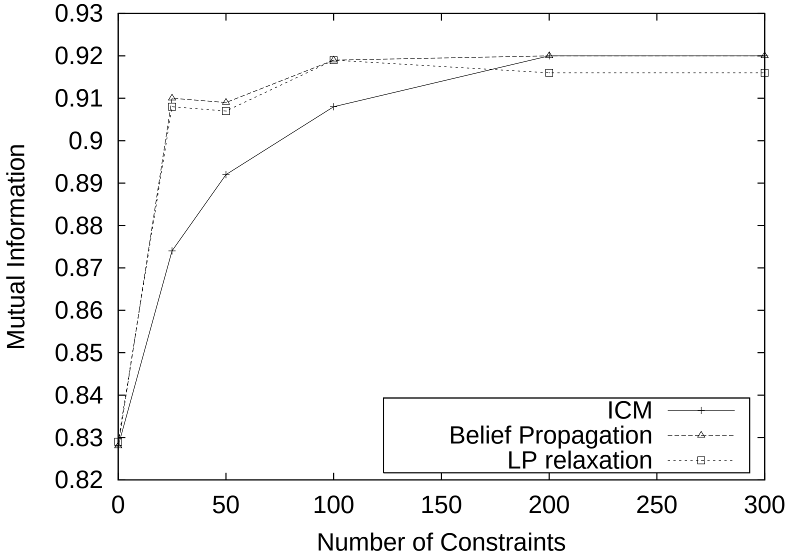

Figure 4.13: Comparison of ICM, BP and LP on *Iris* data set

semi-supervised clustering using a broad class of distortion functions, namely *Bregman divergences* (Banerjee et al., 2004), which include a wide variety of useful distances, e.g., KL divergence, squared Euclidean distance, and Itakura-Saito distance. In a number of applications, such as text clustering based on a vector-space model, a directional distance measure based on the cosine of the angle between vectors is more appropriate (Baeza-Yates & Ribeiro-Neto, 1999). Clustering algorithms have been developed that utilize distortion measures appropriate for directional data (Dhillon & Modha, 2001; Banerjee et al., 2003), and the HMRF-KMEANS framework naturally extends them. We also perform experiments on both low-dimensional and high-dimensional data sets to show the effectiveness of the HMRF-KMEANS algorithm. Overall, our results show that the HMRF-KMEANS algorithm effectively incorporates constraints and unlabeled data in both the initialization and assignment stages, each of which improves the clustering quality. We have also shown how ICM, a greedy technique of assigning points to clusters in the E-step of the algorithm, is efficient and comparable in accuracy to more expensive global collective inference techniques.

Chapter 5

Active Learning for Constraint Acquisition

In the semi-supervised setting where training data is not already available, getting constraints on pairs of data points may be expensive. In this chapter, we present an active learning scheme for the HMRF model, which can improve clustering performance with as few queries as possible (Basu, Banerjee, & Mooney, 2004). In order to get pairwise constraints that are more informative than random in the HMRF model, we develop a 2-phase active learning scheme for selecting pairwise constraints by asking queries an interactive user-driven semi-supervised clustering framework.

5.1 Problem definition

Formally, the active learning scheme has access to a (noiseless) oracle – the user. The algorithm can pose a constant number of pairwise queries to the oracle, wanting to know the type of constraint on a given pair of instances (x_i, x_j). The oracle can assign a must-link or cannot-link to a given pair; the oracle can also give a *don't-know* response to a query, in which case that response is ignored (the pair is not considered as a constraint) and that query is not posed again later. The goal is to ask the minimal number of queries to get constraints, which, when used to cluster the data with HMRF-KMEANS, will give a better constrained clustering of the data than that obtained using randomly chosen constraints.

The motivation for using our active learning algorithm for selecting good constraints is as follows. In Sec. 3.3.2, it was observed that initializing KMeans with centroids estimated from a set of labeled examples for each cluster gives significant performance improvements. Since good initial centroids are very critical for the success of greedy algorithms such as KMeans, the same principle is followed for the pairwise case: the goal in active learning is to get as many points as possible per cluster (proportional to the actual cluster size) by asking pairwise queries, so that HMRF-KMEANS is initialized from a very good set of centroids. A similar argument can be used to motivate the active learning algorithm for other non-Gaussian exponential distributions.

The proposed active learning scheme has two phases: EXPLORE and CONSOLIDATE, which are discussed in detail in the following sections.

5.2 Exploration

```
Algorithm: EXPLORE
Input: Set of data points $X = \{x_i\}_{i=1}^{n}$, access to an oracle that answers pairwise
    queries, number of clusters $k$, total number of queries $Q$.
Output: $\lambda \leq k$ disjoint neighborhoods $N = \{N_p\}_{p=1}^{\lambda}$ corresponding to the true
    clustering of $X$ with at least one point per neighborhood.
Method:
1. Initialize: set all neighborhoods $N_p$ to null
2. Pick the first point $x$ at random, add to $N_1$, $\lambda \leftarrow 1$
3. While queries are allowed and $\lambda < k$
        $x \leftarrow$point farthest from the points in the existing neighborhoods $N$
        if, while pairing $x$ with a point from each existing neighborhood and querying,
        it is found that $x$ is cannot-linked to all existing neighborhoods
            $\lambda \leftarrow \lambda + 1$, start a new neighborhood $N_\lambda$ with $x$
        else
            add $x$ to the neighborhood with which it is must-linked
```

Figure 5.1: Explore algorithm

The EXPLORE (Fig. 5.1) phase explores the given data using farthest-first traversal to get k pairwise disjoint non-null neighborhoods as fast as possible, with each neighborhood belonging to a different cluster in the underlying clustering of the data. Note that even if there is only one point per neighborhood, this neighborhood structure defines a correct skeleton of the underlying clustering.

The basic idea of farthest-first traversal of a set of points is to find k points such that they are far from each other. In farthest-first traversal, a starting point is first selected at random. Then, the next point farthest from it is chosen and added to the traversed set. After that, the next point farthest from the traversed set (using the standard notion of distance from a set: $d(x,S) = \min_{x' \in S} d(x,x')$) is selected, and so on. Farthest-first traversal gives an efficient approximation of the *k-center* problem (Hochbaum & Shmoys, 1985), and has also been used to construct hierarchical clusterings with performance guarantees at each level of the hierarchy (Dasgupta, 2002).

In EXPLORE, while queries are still allowed and k pairwise disjoint neighborhoods have not yet been found, the point x farthest from all the existing neighborhoods is chosen as a candidate for starting a new neighborhood. Queries are posed by pairing x with an arbitrary point from each of the existing neighborhoods. If x is cannot-linked to all the existing neighborhoods, a new neighborhood is started with x. If a must-link is obtained for a particular neighborhood, x is added to that neighborhood. This continues till the algorithm runs out of queries or k pairwise disjoint neighborhoods have been found. In the latter case, active learning enters the consolidation phase.

```
Algorithm: CONSOLIDATE
Input: Set of data points $X = \{x_i\}_{i=1}^n$, access to an oracle that answers pairwise
    queries, number of clusters $k$, total number of queries $Q$, $k$ disjoint neighborhoods
    corresponding to true clustering of $X$ with at least one point per neighborhood.
Output: $k$ disjoint neighborhoods corresponding to the true clustering of $X$ with
    higher number of points per neighborhood.
Method:
1. Estimate centroids $\{\mu_h\}_{h=1}^k$ of each of the neighborhoods
2. While queries are allowed
2a.   randomly pick a point $x$ not in the existing neighborhoods
2b.   sort the indices $h$ with increasing distances $||x - \mu_h||^2$
2c.   for  $h = 1$ to $k$
          query $x$ with each of the neighborhoods in sorted order till a must-link is
          obtained, add $x$ to that neighborhood
```

Figure 5.2: Consolidate algorithm

5.3 Consolidation

If we reach the end of EXPLORE without running of out queries, then at least one point has been obtained per cluster. If there are any remaining queries, they are used to consolidate this structure. The cluster skeleton obtained from EXPLORE is used to initialize k pairwise disjoint non-null neighborhoods $\{N_p\}_{p=1}^{k}$. Then, given any point x not in any of the existing neighborhoods, we will have to ask at most $(k-1)$ queries by pairing x up with a member from each of the disjoint neighborhoods N_p to find out the neighborhood to which x belongs. This principle forms the second phase of our active learning algorithm, and we call the algorithm for this phase CONSOLIDATE. In this phase, we are able to get the correct cluster label of x by asking at most $(k-1)$ queries.

The consolidation phase starts when at least one point has been obtained from each of the k clusters. The basic idea in CONSOLIDATE (Fig. 5.2) is as follows: since there is at least one labeled point from all of the clusters, the proper neighborhood of any unlabeled point x can be determined within a maximum of $(k-1)$ queries. The queries will be formed by taking a point y from each of the neighborhoods in turn and asking for the label on the pair (x,y) until a must-link is obtained. Either a must-link reply is obtained in $(k-1)$ queries, or if we get cannot-link replies for the $(k-1)$ queries to the $(k-1)$ neighborhoods, we can infer that the point is must-linked to the remaining neighborhood. Note that it is practical to sort the neighborhoods in increasing order of the distance of their centroids from x so that the correct must-link neighborhood for x is encountered sooner in the querying process.

5.4 Motivation of EXPLORE Vs CONSOLIDATE

Our exploration phase is motivated by a property of the farthest-first traversal, applicable to all bounded symmetric distance functions $d(x,y)$. Considering 2 disjoint balls, defined in terms of the distance function, of uniform probability density (see Appendix B.1). The balls are of unequal size, implying unequal probability mass. If the ratio of the probability mass of the smaller to the larger ball is lower bounded by $\frac{1}{\ell}$ for a positive integer ℓ, then the farthest-first scheme is sure to get one point from each of the balls in at most ℓ traversals (see Appendix B.3). Motivated by this property, EXPLORE uses farthest-first traversal for getting a skeleton structure of the neighborhoods, and terminates when it has run out of queries, or, when at least one point from all the clusters has been labeled.

Both EXPLORE and CONSOLIDATE add points to the clusters at a good rate. The EXPLORE phase gets at least one point from each of the k underlying clusters in maximum $k\binom{k}{2}$ queries, while CONSOLIDATE gets one new point from each cluster in approximately $k^2 \log k$ queries with high probability (see Appendix B.3). CONSOLIDATE therefore adds points to clusters at a faster rate than EXPLORE by a factor of $O(\frac{k}{\log k})$, which is validated by our experiments in Sec. 5.5. Note that this analysis is for balanced clusters, but a similar analysis with unbalanced clusters gives the same improvement factor.

When the right number of clusters k is not known to the clustering algorithm, k is also unknown to the active learning scheme. In this case, only EXPLORE is used while queries are allowed. EXPLORE will keep discovering new clusters as fast as it can. When it has obtained all the clusters, it will not have any way of knowing this. However, from this point onwards, for every farthest-first x it draws from the data set, it will always find a neighborhood that is must-linked to it. Hence, after discovering all of the clusters, EXPLORE will essentially consolidate the clusters too. However, when k is known, it makes sense to invoke CONSOLIDATE since (1) it adds points to clusters at a faster rate than EXPLORE, and (2) it picks random samples following the underlying data distribution, which is advantageous for estimating good centroids (e.g., Chernoff bounds on the centroid estimates exist, as shown in Eqn. (3.4)), while samples obtained using farthest-first traversal may not have such properties.

5.5 Experiments

In this section, we outline the details of our experiments on text and UCI data and analyze the results.

5.5.1 Data sets

In our experiments with high-dimensional text documents, we used the 3 small subsets of *20-Newsgroups-1000* described in Sec. 4.5.1, and *20-Newsgroups-100*, which was described in Sec. 3.5.1. Another data set we used in our experiments is a subset of *Clas-*

sic3 (Dhillon & Modha, 2001) containing 400 documents – 100 `Cranfield` documents from aeronautical system papers, 100 `Medline` documents from medical journals, and 200 `Cisi` documents from information retrieval papers. This *Classic3-subset* data set was specifically designed to create clusters of unequal size, and has 400 points in 2897 dimensions. Similarities between data points in the text data sets were computed using cosine similarity, and all the text data sets were pre-processed following the methodology outlined in Sec. 2.5.

For experiments on low-dimensional data, we selected the *Iris* data set described in Sec. 3.5.1. The Euclidean metric was used for computing distances between points in this data set. The *Iris* data set was not pre-processed in any way.

5.5.2 Methodology

For all of the algorithms, on each data set, we generated learning curves with 10-fold cross-validation, where the x-axis represents the number of pairwise constraints given as input to the algorithms. For non-active HMRF-KMEANS the pairwise constraints are selected at random, while for active HMRF-KMEANS the pairwise constraints are selected using our active learning scheme. For studying the effect of pairwise constraints and active learning, 10% of the data set is set aside as the test set at any particular fold. The training sets at different points of the learning curve are pairwise constraints obtained from the remaining 90% of the data, with increasing number of pairwise constraints being given as input to the clustering along the learning curve. The clustering algorithm is run on the whole data set, and the corresponding objective function is reported. NMI and pairwise F-measure (see Sec. 2.4) are calculated only on the test set, from which no constraints were supplied. We also show results for the objective function $\mathcal{J}_{\text{hmrf-kmeans}}$. The results at each point on the learning curve are obtained by averaging over 10 folds. We did not continue the learning curve beyond 1000 queries (5000 for *20-Newsgroups-100*), since the general nature of the results was evident in this range. Moreover, in practical active learning applications, it is unrealistic to expect the user to answer even 1000 queries.

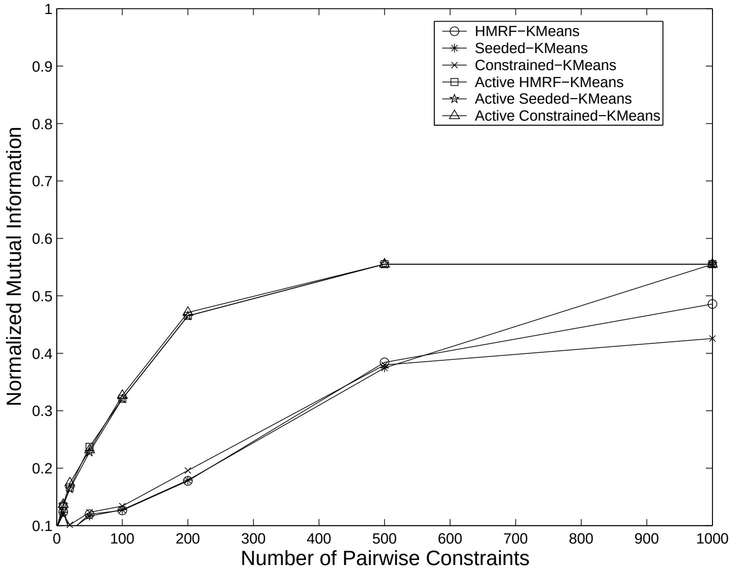

Figure 5.3: Comparison of NMI values on *3-News-Similar-100*

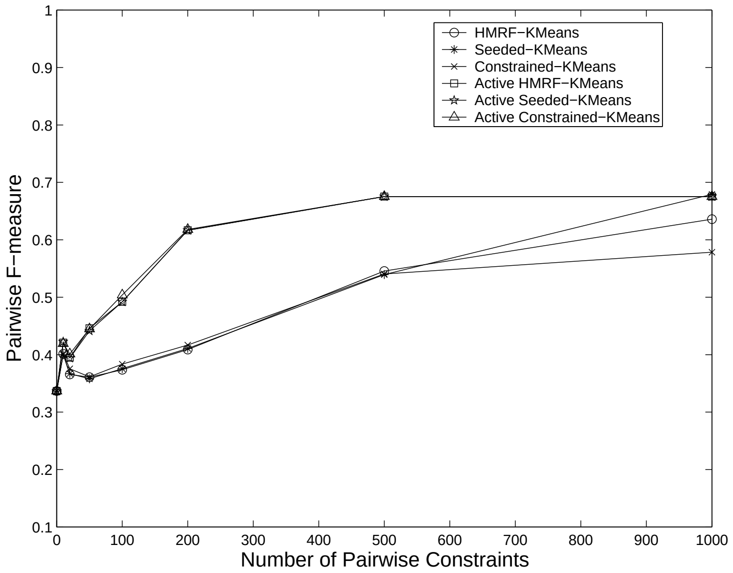

Figure 5.4: Comparison of pairwise F-measure values on *3-News-Similar-100*

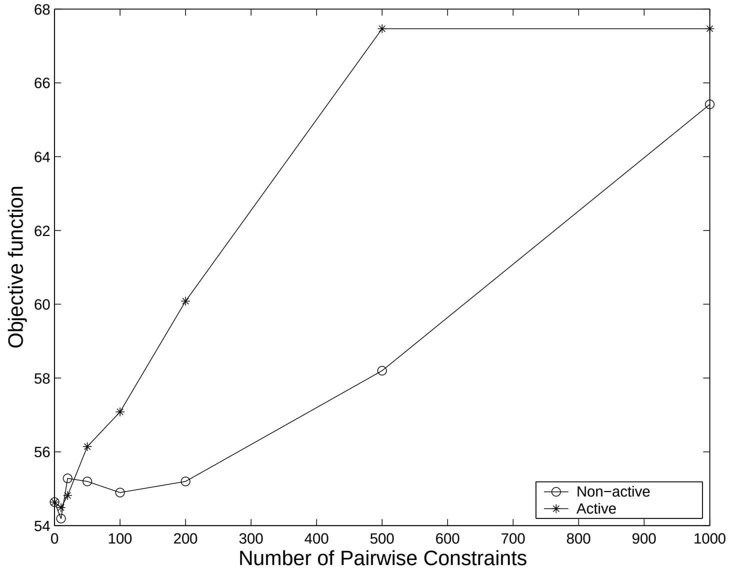

Figure 5.5: Comparison of objective function on *3-News-Similar-100*

5.5.3 Results and discussion

The results of the experiments are shown in Figs. 5.3-5.17. Since the standard deviations of NMI, pairwise F-measure and objective function values in the plots were small for all the data sets, they have not been shown in the plots to reduce clutter.

Choice of *w*: We experimented with different values of the constraint weight parameter *w*. If *w* is set to 0, the algorithm is initialized with neighborhoods derived from the given constraints and then normal KMeans iterations are run till convergence. This is similar to the SEEDED-KMEANS algorithm outlined in Sec. 3.2, where the labeled data (seeds) are used to only initialize the KMeans algorithm and are not used in the following steps of the algorithm.

If *w* is set to a very high value, the algorithm is initialized with neighborhoods derived from the given constraints and the constraints become hard constraints, since the constraint cost violation component of the $\mathcal{J}_{\text{hmrf-kmeans}}$ objective function far supersedes its distance component. This is similar to the CONSTRAINED-KMEANS algorithm outlined in Sec. 3.2. In this algorithm, the seeds are also used to initialize the KMeans algorithm. However, in the subsequent steps, the cluster labels of the seed data are kept unchanged and only the labels of the non-seed data are re-estimated.

If *w* is set to an intermediate value, the algorithm gives a tradeoff between minimizing the total distance between points and cluster centroids and the cost of violating the constraints. In the result plots in Figs. 5.3 and 5.4, HMRF-KMEANS refers to running the algorithm with the intermediate value of *w*. The parameter *w* can be chosen by the user according to the degree of confidence in the constraints, or chosen to be a constant of the same order as the average similarity (for SP-KMeans) or distance (for Euclidean KMeans) between pairs of points in the data set. We set *w* to be 0.001 for the text data sets and 1 for *Iris* data set.

Thus, the *w* parameter acts as a tuning knob, giving us the continuum between a SEEDED-KMEANS-like algorithm on one extreme, where there is no guarantee of the constraint satisfaction in the clustering, and a CONSTRAINED-KMEANS-like algorithm on the other extreme, where the clustering process is forced to respect all the given constraints. Note that we can selectively guarantee that any particular constraint is satisfied throughout the clustering iterations, by selecting a very high corresponding cost of constraint violation for that particular constraint.

The comparative results of active and non-active algorithms obtained for different values of *w* were similar for the data sets considered (see Figs. 5.3 and 5.4). This leads us to conclude that proper initialization, using the constraints obtained by active learning, gives much more benefit than satisfying the constraints during the algorithm. This point is explained in more detail in the discussion below. In Figs. 5.6-5.17, we only present the results for the intermediate value of *w* for clarity of the plots.

Objective function results: Let us consider a representative objective function plot for a text data set clustered using SP-KMeans (Fig. 5.8), for which the objective function increases along the learning curve. For Fig. 5.17, the objective function is decreasing along

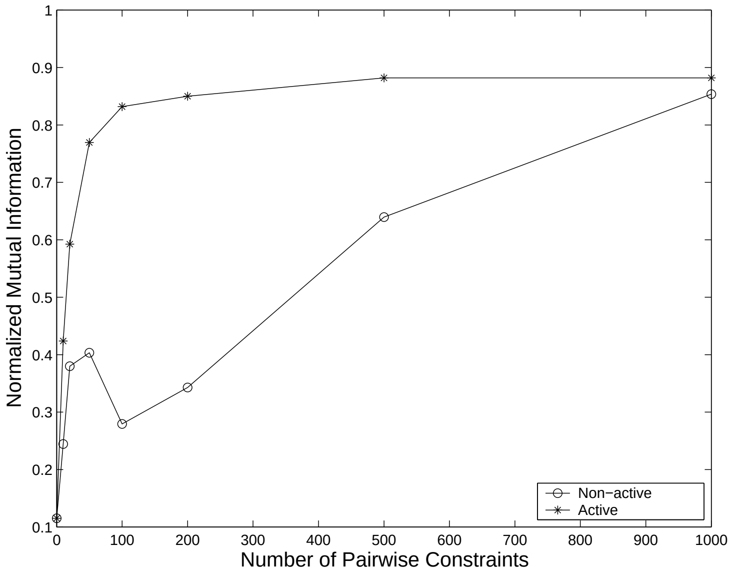

Figure 5.6: Comparison of NMI values on *3-News-Different-100*

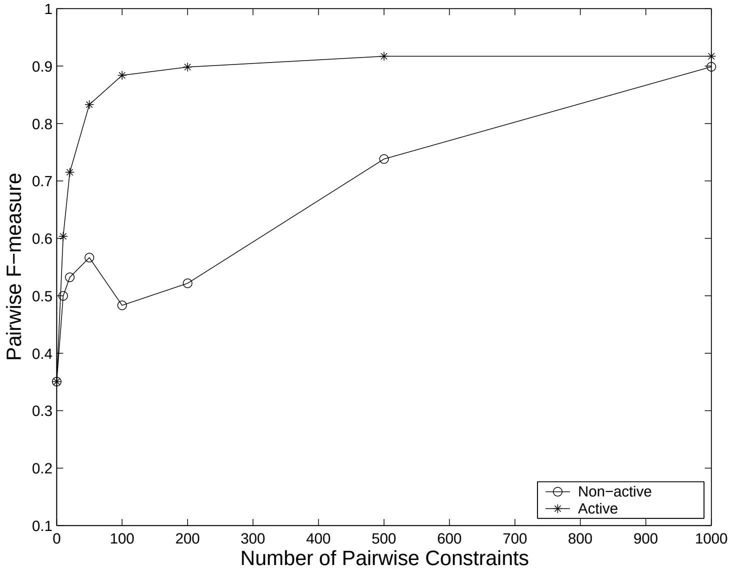

Figure 5.7: Comparison of pairwise F-measure values on *3-News-Different-100*

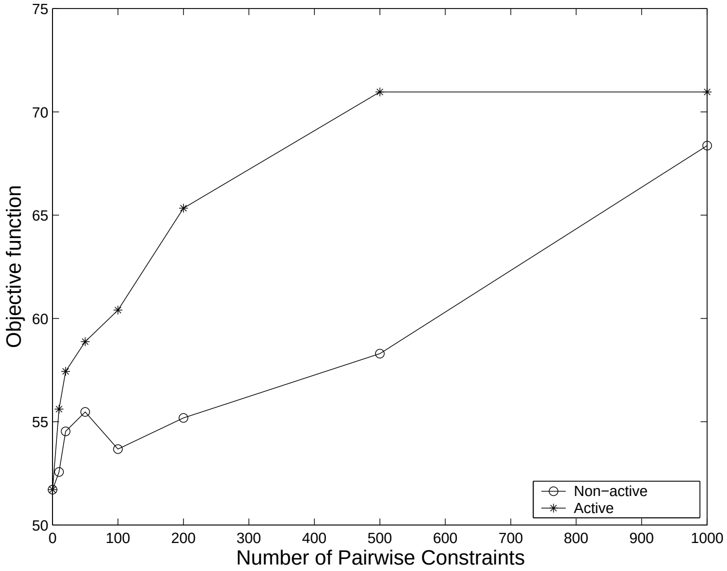

Figure 5.8: Comparison of objective function on *3-News-Different-100*

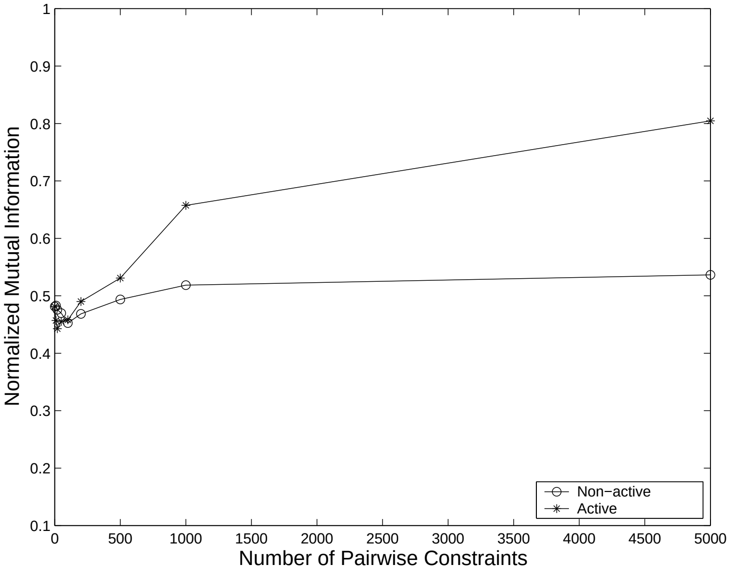

Figure 5.9: Comparison of NMI values on *20-Newsgroups-100*

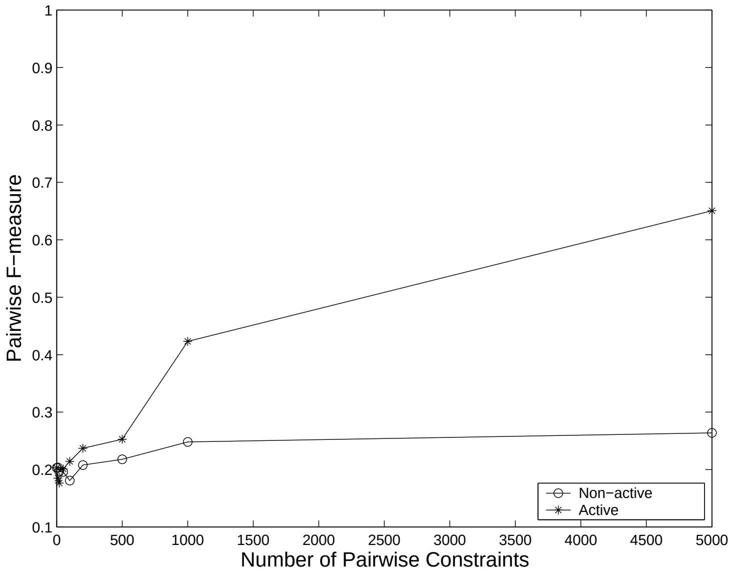

Figure 5.10: Comparison of pairwise F-measure values on *20-Newsgroups-100*

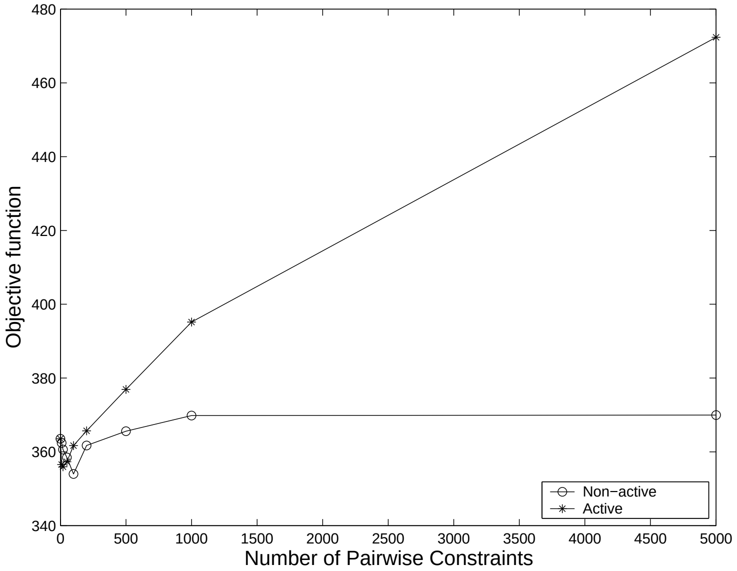

Figure 5.11: Comparison of objective function on *20-Newsgroups-100*

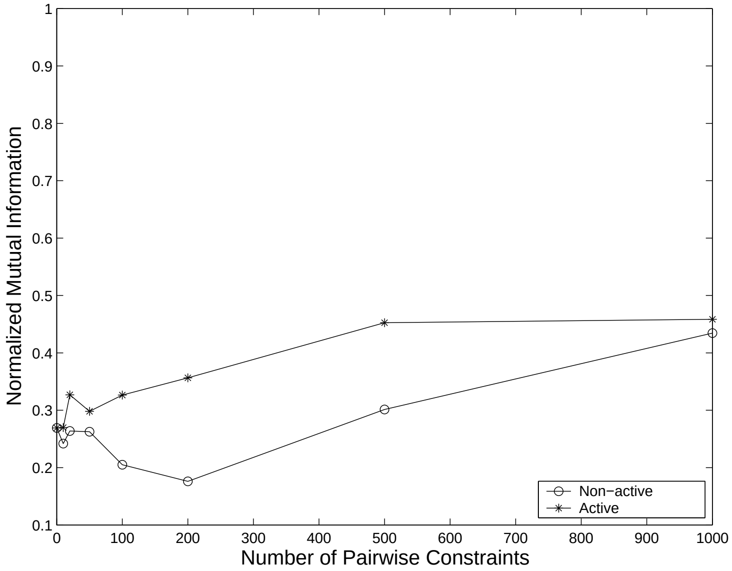

Figure 5.12: Comparison of NMI values on *Classic3-subset*

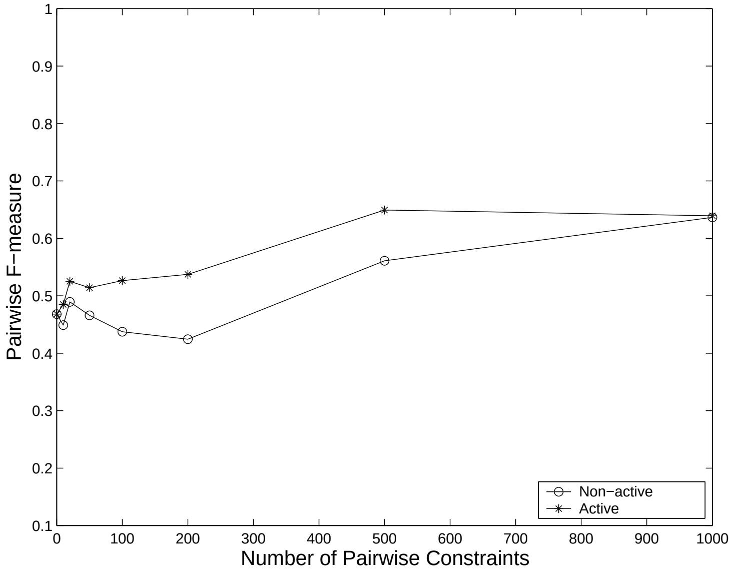

Figure 5.13: Comparison of pairwise F-measure values on *Classic3-subset*

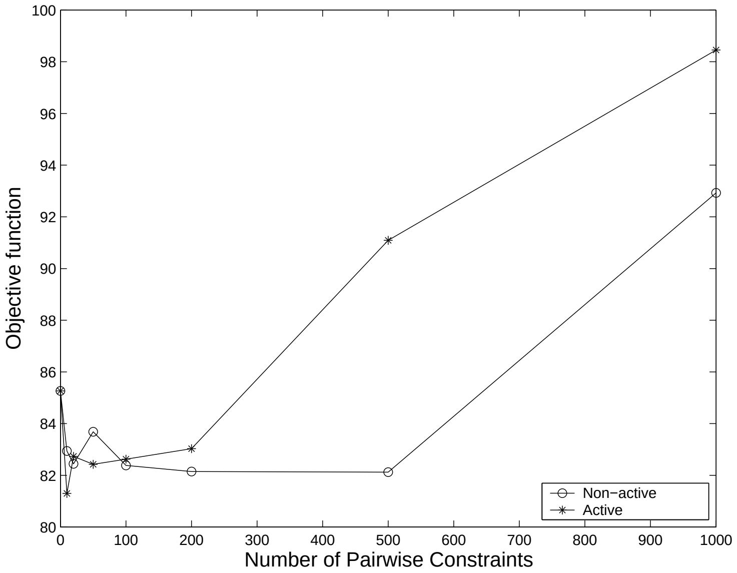

Figure 5.14: Comparison of objective function on *Classic3-subset*

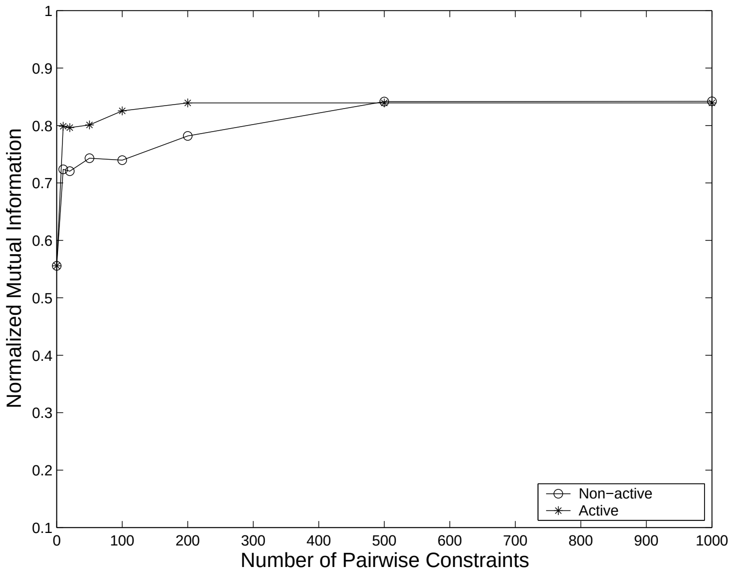

Figure 5.15: Comparison of NMI values on *Iris*

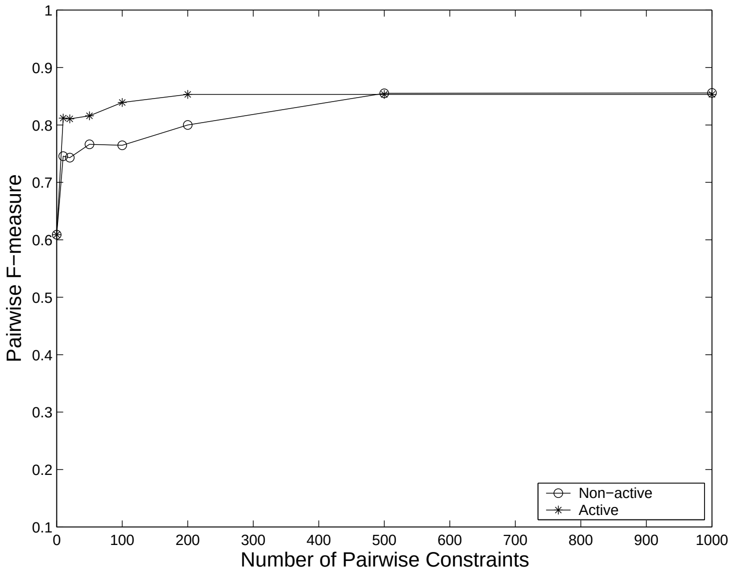

Figure 5.16: Comparison of pairwise F-measure values on *Iris*

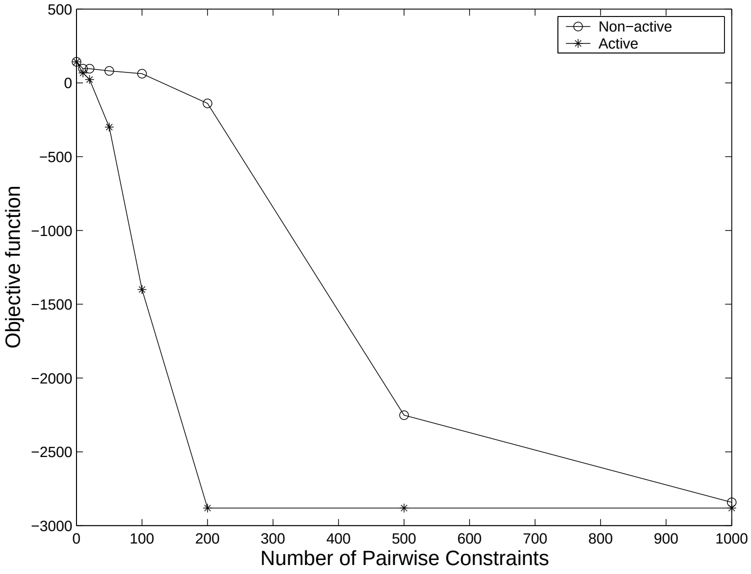

Figure 5.17: Comparison of objective function on *Iris*

the learning curve since simple KMeans with Euclidean distance was used for this data set.

Note that for each objective function plot, the active and non-active schemes have the same number of must-link and cannot-link constraints at any point on the learning curve, but the actual constraints they have may be different. The active and the non-active schemes are allowed to both choose their own sets of constraints, and the objective function value after running HMRF-KMEANS clustering depends on this choice. For active HMRF-KMEANS, the constraints it chooses give it a better initialization (which is discussed in detail below), resulting in better value of the objective function after running the clustering algorithm.

Non-active schemes: As shown in Appendix B.2, if the number of random pairwise constraints is low, the probability that the *k* neighborhoods chosen for initialization are in fact from *k* different clusters is very low. Until this point on the learning curve, some of the neighborhoods used to initialize HMRF-KMEANS can actually belong to the same cluster, so that we may not get representatives from all the clusters. This gives a poor initialization of HMRF-KMEANS that may cause the algorithm to converge to bad local minima. Consequently, the clustering produced by HMRF-KMEANS can be unstable, resulting in varying pairwise F-measure and NMI values on the test set. This initial jitter can be observed in all the Figs. 5.3-5.17. Beyond this point on the learning curve, non-active HMRF-KMEANS will most likely be initialized with points from each cluster. So after the initial jitter, the performance of non-active HMRF-KMEANS improves steadily along the learning curve with respect to objective function, NMI and pairwise F-measure.

Active schemes: For the active algorithms, we consistently get significant improvements over the non-active algorithms for all data sets we have considered. Firstly, we see the jitter only in the very early part of the learning curve. This is because the EXPLORE phase creates only one neighborhood from each cluster and continues until *k* pairwise disjoint neighborhoods are found, creating all the neighborhoods within a small number of queries (see Appendix B.3). The jitter is so early in the learning curve that it cannot be even observed in the plots. In Fig. 5.9, the jitter disappears after about the first 20 queries. The EXPLORE phase of the active selection algorithm guarantees that the pairwise disjoint neighborhoods inferred from the constraints belong to different clusters in the actual underlying clustering, and so these neighborhoods would give us good initializations for the clustering algorithm. The CONSOLIDATE phase grows the *k* pairwise disjoint neighborhoods already created, so that when the active learning scheme runs out of queries, HMRF-KMEANS is initialized using centroids constructed from good neighborhoods. The improvement of the active scheme is more pronounced for the difficult high-dimensional text data sets, e.g., Fig. 5.3-5.14.

From the above results, we conclude that active selection of pairwise constraints, using our two-phase active learning algorithm, significantly outperforms random selection of constraints.

Explore Vs Consolidate: We also ran some ablation experiments, comparing the performance of the active HMRF-KMEANS scheme with both EXPLORE and CONSOLIDATE to active HMRF-KMEANS with EXPLORE only. We ran the ablation experiment on

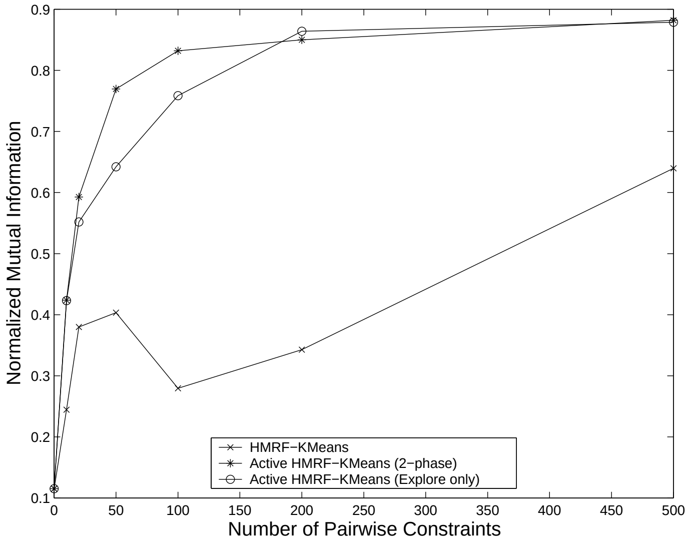

Figure 5.18: Comparison of Explore and Consolidate phases w.r.t. NMI on *3-News-Different-100*

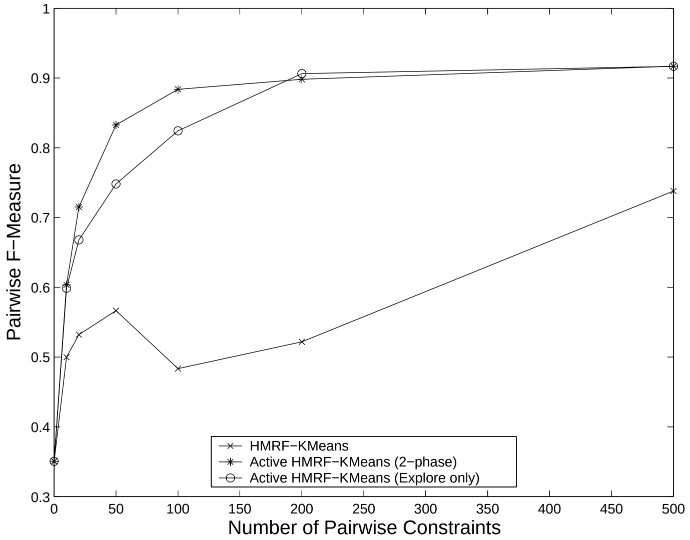

Figure 5.19: Comparison of Explore and Consolidate phases w.r.t. pairwise F-measure on *3-News-Different-100*

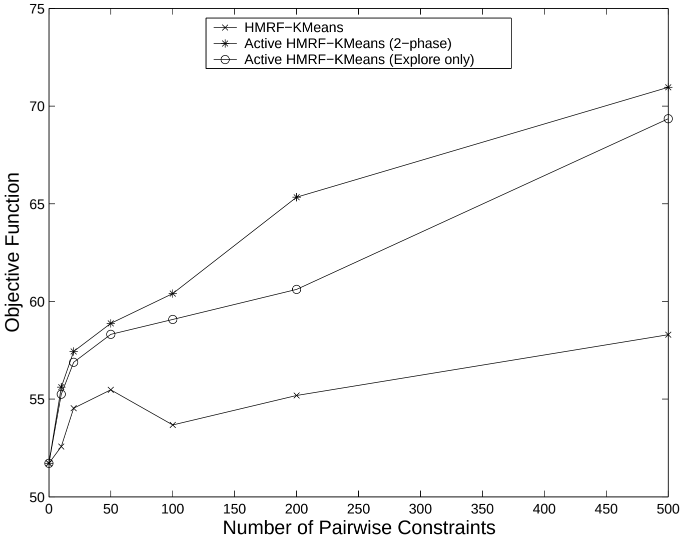

Figure 5.20: Comparison of Explore and Consolidate phases w.r.t. objective function on *3-News-Different-100*

the *3-News-Different-100* data set. From the NMI result shown in Fig. 5.18, we can see that running EXPLORE only in the active learning phase gives improvement over random choice of constraints, but running both EXPLORE and CONSOLIDATE gives even better results. So, both EXPLORE and CONSOLIDATE are useful phases of the active learning algorithm. However when the number of clusters is not known, just using EXPLORE (as recommended in Sec. 5.4) can give pretty good results, as demonstrated by Fig. 5.18.

5.6 Chapter summary

In this chapter, we have presented a new theoretically well-motivated method for actively selecting good pairwise constraints for semi-supervised clustering. Experiments on text and UCI data show that our active learning scheme performs quite well, giving significantly steeper learning curves compared to random pairwise queries. Both phases of the active learning algorithm are efficient and hence suitable for real-world clustering applications, as they can be easily scaled to large and high-dimensional data sets.

Chapter 6

Related Work

Several semi-supervised classification algorithms have shown improvements in classification accuracy over purely supervised algorithms, e.g., co-training (Blum & Mitchell, 1998), transductive Support Vector Machines (SVMs) (Joachims, 1999), and semi-supervised EM (Ghahramani & Jordan, 1994; Nigam et al., 2000). In contrast, this thesis discusses semi-supervised clustering. The following sections outline current and previous research related to the work presented in this thesis.

6.1 Semi-supervised clustering with labels

In semi-supervised clustering with labeled data, previous work has been done on the use of labeled data to aid clustering by modifying clustering objective functions (Demiriz et al., 1999), and using conditional distributions in an auxiliary space (Sinkkonen & Kaski, 2000).

SEEDED-KMEANS and CONSTRAINED-KMEANS in Chapter 3 use the labeled data to initialize clustering. Previous work on cluster initialization includes comparisons of data-dependent and data-independent initialization techniques (Meila & Heckerman, 1998), and estimation of the modes of the data distribution for good initialization (Fayyad et al., 1998). The importance of good initialization in clustering is well-known. In partitional clustering algorithms like EM (Dempster et al., 1977) or KMeans (MacQueen, 1967; Selim & Ismail, 1984), some commonly used approaches for initialization include simple random selection, taking the mean of the whole data and randomly perturbing to get initial cluster centers (Dhillon et al., 2001), or running *k* smaller clustering problems recursively to initialize KMeans (Duda et al., 2001). Some other interesting initialization methods include the Buckshot method of doing hierarchical clustering on a sample of the data to get an initial set of cluster centers (Cutting, Karger, Pedersen, & Tukey, 1992), running repeated KMeans on multiple data samples and clustering the KMeans solutions to get initial seeds (Fayyad et al., 1998), and selecting the *k* densest intervals along each co-ordinate to get the *k* cluster centers (Bradley, Mangasarian, & Street, 1997). Our approach is different from these because we use labeled data to get good initialization for clustering.

6.2 Semi-supervised clustering with constraints

Previous research in semi-supervised clustering with constraints focus on either constraint-based or distance-based semi-supervised clustering. COP-KMeans is a constraint-based clustering algorithm that has a heuristically motivated objective function (Wagstaff et al., 2001). On the other hand, the model of semi-supervised clustering presented in Chapter 4 has an underlying probabilistic model based on Hidden Markov Random Fields. Bansal et al. (Bansal, Blum, & Chawla, 2002), Blum et al. (Blum, Lafferty, Rwebangira, & Reddy, 2004) and Charikar et al. (Charikar, Guruswami, & Wirth, 2003) also propose frameworks for pairwise constrained clustering, but their model performs clustering using only the constraints; in comparison, HMRF-KMEANS uses both constraints and an underlying distortion measure between the points during semi-supervised clustering.

Research on distance-based semi-supervised clustering with pairwise constraints includes the work of Cohn et al. (Cohn et al., 2003), who used gradient descent for weighted Jensen-Shannon divergence in the context of EM clustering; Xing et al. (Xing et al., 2003) utilized convex optimization and iterative projections to learn a Mahalanobis distance for K-Means clustering; the Redundant Component Analysis (RCA) algorithm used only must-link constraints to learn a Mahalanobis distance using convex optimization (Bar-Hillel et al., 2003). Other methods include training a string-edit distance using Expectation Maximization (EM) (Bilenko & Mooney, 2003), modification of the squared Euclidean distance using the shortest-path algorithm (Klein et al., 2002), learning a margin-based clustering distortion measure using boosting (Hertz et al., 2004), and learning a distance metric transformation that is globally linear but locally non-linear (Chang & Yeung, 2004). Spectral learning (Kamvar, Klein, & Manning, 2003) is another method that utilizes supervision to transform the clustering distance measure using spectral methods. All of these distance-learning techniques for clustering train the distance measure first using only supervised data, and then perform clustering on the unsupervised data. In contrast the unified HMRF-based semi-supervised clustering model, discussed briefly in Chapter 7, integrates distance learning with the clustering process, and utilizes both supervised and unsupervised data to learn the distortion measure.

A model for semi-supervised clustering with constraints was proposed by Segal et al. (Segal et al., 2003a). This model is a *Markov network* that combines a binary Markov network derived from pairwise protein interaction data and a Naive Bayes Markov network modeling gene expression data. The HMRF framework proposed in this thesis generalizes that formulation by extending it to work with a broad class of clustering distortion measures, including Bregman divergences and cosine distance. In comparison, the formulation of Segal et al. considers only a Gaussian cluster conditional probability distribution, which corresponds to having Mahalanobis distance as the underlying clustering distance measure.

The HMRF-KMEANS algorithm is related to the EM algorithm for HMRF model-fitting proposed by Zhang et al. (Zhang et al., 2001). The discussion of the HMRF-EM algorithm was also restricted only to Gaussian conditional distributions, which has been generalized in HMRF-KMEANS. Other recent research on constrained clustering includes variational techniques for constrained clustering using a graphical model (Hiu et al., 2005), model-level constraints to uncover multiple constraints in a dataset (Gondek, Vaithyanathan, & Garg, 2005), and feasibility studies for clustering under different types of constraints (Davidson & Ravi, 2005).

6.3 Active learning for constraint acquisition

Active learning in the classification framework is a long-studied problem, where different principles of query selection have been studied, e.g., reduction of the version space size (Freund, Seung, Shamir, & Tishby, 1997), reduction of uncertainty in predicted label (Lewis & Gale, 1994), maximizing the margin on training data (Abe & Mamitsuka, 1998), finding high variance data points by density-weighted pool-based sampling (McCallum & Nigam, 1998), etc. However, active learning techniques in classification are not applicable in the clustering framework, since the basic underlying concept of reduction of classification error and variance over the distribution of examples (Cohn, Ghahramani, & Jordan, 1996) is not well-defined for clustering. In the unsupervised setting, Hofmann et al. (Hofmann & Buhmann, 1998) consider a model of active learning which is different from ours – they have incomplete pairwise similarities between points, and their active learning goal is to select new data, using expected value of information estimated from the existing data, such that the risk of making wrong estimates about the true underlying clustering from the existing incomplete data is minimized. In contrast, our model assumes that we have complete similarity information between all pairs of points and pairwise constraints whose violation cost is a component of the objective function, and the active learning goal is to select pairwise constraints which are most informative about the underlying clustering. Klein et al. (Klein et al., 2002) also consider active learning in semi-supervised clustering, but instead of making example-level queries they make cluster level queries, i.e., they ask the user whether or not two whole clusters should be merged. Answering example-level queries rather than cluster-level queries is a much easier task for a user, making our model more practical in a real-world active learning setting.

Chapter 7

Other Results in Semi-supervised Clustering

In this chapter, we present some interesting problems related to semi-supervised clustering that have not been discussed so far in this thesis and present some ideas of future research in some of these areas. Most of the work presented in this chapter was done in collaboration with other researchers at the University of Texas at Austin.

7.1 Unified model for constrained semi-supervised clustering

We developed a generalization of HMRF-KMEANS that incorporates *both* distortion measure learning and the use of pairwise constraints in a principled manner (Basu et al., 2004). This was done by using parameterized distortion measures that can be adapted to specific datasets: in this method, the distortion measure parameters are updated in the M-step of the algorithm during the clustering iterations, in order to get a learned measure that puts must-linked points closer together and pulls cannot-linked points further apart.

Previous distance-based semi-supervised clustering algorithms exclude unlabeled data from the distortion measure learning step, as well as separate distance learning from the clustering process (see Sec. 1.3.2). Also, existing distance-based methods use a single distance metric for all clusters, forcing them to have similar shapes. The unified HMRF-KMEANS algorithm is able to perform distance learning with each clustering iteration, utilizing both unlabeled data and pairwise constraints. It allows violation of constraints if it leads to a more cohesive clustering, whereas earlier constraint-based methods forced satisfaction of all constraints, leaving them vulnerable to noisy supervision. The algorithm is also able to learn individual distortion measures for each cluster, which permits clusters of different shapes.

7.2 Semi-supervised graph-based clustering

Since pairwise constraints are a natural form of supervision for data sets represented in the form of a graph, an interesting problem in clustering is the study of how to incorporate pairwise constraints into a graph clustering (a.k.a. graph partitioning) algorithm, wherein the nodes of the graph are partitioned into sets based on some objective criterion defined over the graph edges (Chan, Schlag, & Zien, 1994; Shi & Malik, 2000). We have recently proposed a semi-supervised clustering algorithm that can work on both vector-based and graph-based data sets (Kulis, Basu, Dhillon, & Mooney, 2005). In this work, we use a recent theoretical connection between kernel *k*-means and several graph clustering objectives, which enables us to perform semi-supervised clustering of data given either as vectors or as a graph. For vector data, our approach generalizes the HMRF-KMEANS algorithm for squared Euclidean distance to work with kernels, which enables it to find clusters with non-linear boundaries in the input data space. For graph data, we show that recent work on spectral learning (Kamvar et al., 2003) may be viewed as a special case of our formulation.

This result currently shows the connection between spectral objective functions for graph partitioning and the corresponding vector-based clustering using only squared Euclidean distance as the clustering distortion measure for the vector data. In the future, we want to extend this and show the equivalence between spectral clustering and kernel-based KMeans clustering for any regular Bregman divergence defined between the input vectors.

7.3 Semi-supervised overlapping clustering

While the vast majority of clustering algorithms are partitional, many real world datasets have inherently overlapping clusters. The recent explosion of analysis on biological datasets, which are frequently overlapping, has led to new clustering models that allow hard assignment of data points to multiple clusters. One particularly appealing model was proposed by Segal et al. (Segal, Battle, & Koller, 2003b) in the context of probabilistic relational models (PRMs) applied to the analysis of gene microarray data. In recent work with other researchers at the University of Texas at Austin, we started with the basic approach of Segal et al. and proposed an alternative interpretation of the model as a generalization of mixture models, which makes it easily interpretable (Banerjee, Krumpelman, Basu, Mooney, & Ghosh, 2005). While the original model maximized likelihood over constant variance Gaussians, we generalize it to work with any regular exponential family distribution, and corresponding Bregman divergences, thereby making the model applicable for a wide variety of clustering distance functions, e.g., KL-divergence, Itakura-Saito distance, I-divergence. The general model is applicable to several domains, including high-dimensional sparse domains, such as text and recommender systems. We additionally offer several algorithmic modifications that improve both the performance and applicability of the model.

An interesting problem to consider in the case of overlapping clustering is how to handle prior knowledge, e.g., pairwise interactions in the Database of Interacting Proteins (DIP) can be used as constraints while performing overlapping clustering of gene data sets. Moreover, the background knowledge in certain domains (e.g., biology) are available from multiple heterogeneous sources with varying degrees of coverage and noise, which have to be integrated using a robust algorithm. We want to investigate both these problems in our future work.

7.4 Model selection in semi-supervised clustering

The HMRF model also assumes that the right number of clusters is given as an input – in the future, we want to select the number of clusters automatically by incorporating a model selection criterion into the HMRF objective function. Several model selection criteria exist in the literature for selecting the right number of clusters. Criteria like Minimum Description Length (Rissanen, 1978), Bayesian Information Criterion (Pelleg & Moore, 2000) or Minimum Message Length (Wallace & Lowe, 1999) encode the Occam's Razor principle in some form, penalizing models according to their model complexity. These criteria can be directly incorporated into the HMRF-KMEANS objective function.

Another interesting model selection technique for clustering that we want to investigate is the PAC-MDL method (Banerjee & Langford, 2004). The PAC-MDL method defines a prediction accuracy model from a clustering; it then trades off between the accuracy of the clustering prediction on the provided labeled (training) data versus the model description length of the clustering, with the goal of getting better prediction accuracy on future unknown (test) data. We have some ideas on how to extend the PAC-MDL model to work with supervision provided in the form of constraints instead of labeled data, which can then be naturally applied to the HMRF-KMEANS model of constrained clustering.

Chapter 8

Future Work

In this chapter, we outline potential future work related to the problems that we discussed in Chapters 3, 4 and 5 of this thesis.

8.1 Label-based semi-supervised clustering

In semi-supervised classification, all classes are assumed to be known a priori and labeled training data is provided for all classes. In labeled semi-supervised clustering, when we consider clustering a dataset that has an underlying class labeling, we would like to consider incomplete seeding – where labeled data are not provided for every underlying class. For such incomplete semi-supervision, we would like to see if the labels on some classes can help the clustering algorithm discover the unknown classes. An example of class discovery using incomplete seeding is provided in the Fig. 8.1. Given the points in Fig. 8.1, if we are asked to do a 2-clustering, we can get a clustering as shown in Fig. 8.1. Now, if we give as input a pair of points labeled to be in the same cluster (shown by the annular points in Fig. 8.2), we will get a clustering as in Fig. 8.2. In this case, even though we did not provide any supervision about the top cluster, clustering using the provided supervision helped us to discover that cluster.

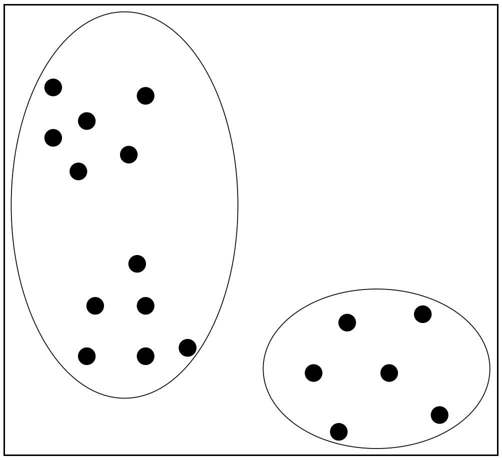

Figure 8.1: Clustering of a sample data set into 2 clusters

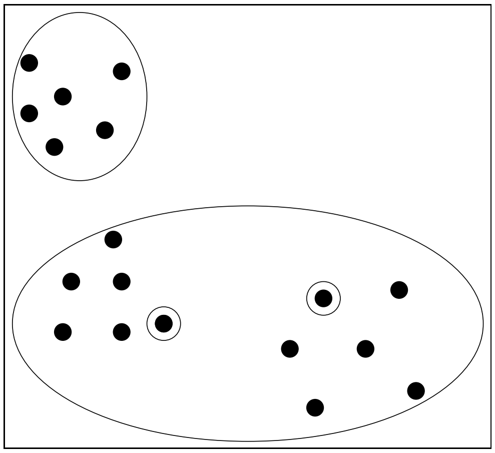

Figure 8.2: Incomplete seeding and class discovery

Initial experiments for class discovery under incomplete seeding were considered in Sec. 3.5, where seeds were not provided for different categories and the NMI measure was calculated on the whole test dataset. In the future, we want to perform more detailed experiments on real domains (e.g., biology) where incomplete supervision is present, with the expectation that in these tasks the semi-supervised clustering algorithm will be able to discover the categories for which no supervision was provided. We also want to come up with a theoretically well-motivated model for class discovery for semi-supervised clustering with labels, similar to the work of Miller et al. (Miller & Browning, 2003).

8.2 Constraint-based semi-supervised clustering

Some aspects of our current clustering model (e.g., initialization in HMRF-KMEANS, EXPLORE phase in active learning) assume that the constraints are consistent, i.e., there is no noise in the constraints. An interesting area of future work would be on incorporating a noise model into our HMRF framework, so that it is able to handle noisy constraints. This would involve some changes to the algorithm, e.g., not adding the inferred constraints between neighborhoods in the initialization step of HMRF-KMEANS, selectively rejecting points using a noise model in the EXPLORE stage of the active learning algorithm, etc.

On a different note, using constraints as supervision has been lately studied in the context of both discriminative classification (Kumar & Hebert, 2003; Yan, Zhang, Yang, & Hauptmann, 2004) and discriminative clustering (Xu, Neufeld, Larson, & Schuurmans, 2005). We want to explore the possibility of training discriminative graphical models for semi-supervised clustering for getting better clustering accuracy.

8.3 Active learning for semi-supervised clustering

The EXPLORE stage of the active learning scheme is currently sensitive to outliers in the data, since the farthest-first traversal can select outliers in the data that do not give much information about the underlying cluster structure, thereby wasting queries during the active learning process. Outlier sensitivity can be handled by density-weighted point selection in EXPLORE, where we could have a modified farthest-first traversal that selects distant points from dense regions of the data space (McCallum & Nigam, 1998). Such a formulation of active learning would be more robust to outliers, and can be used with more outlier-robust clustering algorithms, e.g., KMedian (Mirchandani & Francis, 1990).

Chapter 9

Conclusions

In this thesis, the focus of our research was on semi-supervised clustering, where we study how prior knowledge, gathered either from automated information sources or human supervision, can be incorporated into clustering algorithms. We presented probabilistic models for semi-supervised clustering, developed algorithms based on these models and empirically validated their performances by extensive experiments on data sets from different domains, e.g., text analysis, hand-written character recognition, and bioinformatics.

We proposed a methodology for incorporating supervision in the form of labeled data into clustering using a well-defined EM framework. The two proposed algorithms, SEEDED-KMEANS and CONSTRAINED-KMEANS, use labeled data to form initial clusters and constrain subsequent cluster assignment. Both methods can be viewed as instances of the EM algorithm, where labeled data provides prior information about the conditional distributions of hidden category labels. This interpretation of the semi-supervised clustering algorithms enables us to prove convergence guarantees of both these iterative algorithms. Experimental results clearly demonstrate the advantages of these methods over standard random seeding and COP-KMeans (Wagstaff et al., 2001), an alternative semi-supervised KMeans algorithm. In particular, experiments with simulated noise demonstrated that SEEDED-KMEANS is quite robust to noise in the supervised data.

For supervision provided in the form of pairwise must-link and cannot-link constraints, which are more natural in certain clustering tasks, we proposed a generative probabilistic framework for semi-supervised clustering with constraints. It uses the model of a Hidden Random Markov Field (HMRF) to utilize both unlabeled data and supervision in the form of constraints during the clustering process. The framework is very general and can be used with a wide variety of clustering distortion (distance) measures, including Bregman divergences (e.g., squared Euclidean distance, KL divergence) and directional distances (e.g., cosine distance, Pearson's correlation). We presented an algorithm, HMRF-KMEANS, for performing clustering in this framework – it incorporates supervision in the form of pairwise constraints in both the initialization and cluster assignment stages of the clustering algorithm. In order to demonstrate the effectiveness of each step of the HMRF-KMEANS algorithm, we performed ablation experiments. Particular instantiations of the algorithm gave improved performance for different distortion measures: squared Euclidean distance worked well for clustering low-dimensional UCI data sets, while KL divergence and cosine distance outperformed the individual ablations while clustering high-dimensional directional text data sets.

In a real-life interactive query-driven semi-supervised clustering framework, one challenge is how to acquire pairwise constraints (via queries to the user) that are most helpful to the underlying clustering process. We presented a new active learning method for acquiring supervision from a user in the form of effective pairwise constraints for semi-supervised clustering, which to our knowledge is the first active learning algorithm for constrained clustering. This algorithm has two phases, EXPLORE and CONSOLIDATE, and we empirically demonstrate how both the phases have their utility in the active learning process.

For all the problems mentioned above, we empirically evaluated the effectiveness of our semi-supervised clustering algorithms by detailed experiments on different domains, both low-dimensional (e.g., handwritten character recognition data sets) and high-dimensional (e.g., text documents). Our experiments conclusively demonstrate that using either labeled supervision or pairwise constraints substantially improve the clustering accuracy on different domains, and that our active learning algorithm is able to acquire informative constraints very effectively.

We also discussed other interesting problems of semi-supervised clustering that we studied in collaboration with other researchers, namely (1) integration of both constraint-based and distance-based semi-supervised clustering methods using the HMRF model, (2) semi-supervised graph-based clustering using kernels, (3) using prior knowledge to improve overlapping clustering of data, and (4) model selection techniques that use the available supervision to automatically select the right number of clusters.

Overall, the research presented in this thesis has made significant contributions in theoretically and empirically characterizing semi-supervised clustering, which has become a research topic of significant interest lately. In the general learning setting, the work in this thesis plays an important role in investigating the continuum between completely supervised classification and unsupervised clustering. In the last decade, semi-supervised classification algorithms, which try to improve the performance of classification algorithms using unlabeled data, had been getting considerable attention from machine learning researchers. This thesis takes a different viewpoint of the supervised-unsupervised continuum and looks at another important aspect of semi-supervised learning, namely how to incorporate limited supervision into unsupervised clustering.

The work in this thesis shows how prior knowledge available as labeled data or constraints, which are naturally available in many clustering tasks, can be incorporated into various clustering algorithms. As shown by both theoretical results and empirical evidence, the proposed semi-supervised clustering algorithms give improved performances for various domains, e.g., web search, biometrics, biological data analysis. The research in this thesis would therefore be useful to a large community of clustering practitioners working in different domains. Looking ahead, the algorithms proposed in this thesis and by other researchers working on semi-supervised clustering would become useful tools in the toolboxes of machine learning researchers in the years to come.

Appendix A

Global inference techniques for E-step of HMRF-KMeans

In this appendix, we present two global approximate inference techniques for collective assignment of data points to clusters in the E-step of HMRF-KMeans: belief propagation (BP) and linear programming (LP) relaxation (Basu, Bilenko, & Mooney, 2003).

A.1 Belief propagation approach

A global joint assignment of the points to clusters that (locally) minimizes the objective function $\mathcal{J}_{\text{hmrf-kmeans}}$ can be found by performing approximate inference on the HMRF using belief propagation (Pearl, 1988). This approach is similar to the technique used by Segal et al. (Segal et al., 2003a).

To implement the message passing algorithm for approximate inference on the HMRF, we represent the HMRF as a factor graph model (Kschischang, Frey, & Loeliger, 2001). The sum-product/max-product algorithm on the factor graph model has been shown to be a generalization of several well known inference algorithms on graphical models. Interpreting the HMRF model as a factor graph enables us to perform belief propagation on the HMRF using the max-product message passing algorithm on the corresponding factor graph.

The factor graph corresponding to the example HMRF in Figure 4.1 is shown in Figure A.1. The factor graph has the following components:

(1) n variable nodes $\{x_i\}_{i=1}^{n}$ representing the data points.

(2) n factor nodes $\{D_i\}_{i=1}^{n}$ that encode the distance potential components of the objective function. Each distance factor node D_i has an edge connecting it to the corresponding variable node x_i, and a table containing different values of the distance potential function. This table has an entry for each possible cluster assignment of the variable; the j^{th} entry is $\exp(-d)$, where d is the distance from the i^{th} point to the j^{th} cluster.

(3) $|C_{ML}|$ factor nodes $\{M_i\}_{i=1}^{|C_{ML}|}$ and $|C_{CL}|$ factor nodes $\{C_i\}_{i=1}^{|C_{CL}|}$, which respectively encode the cost of violating the must-link and cannot-link constraints. There is one factor node for each constraint, which is linked by edges to the 2 variable nodes involved in that constraint.

The constraint potential table associated with each constraint factor node maps a set of k^2 value-pairs (corresponding to possible cluster assignments to the pair of points in

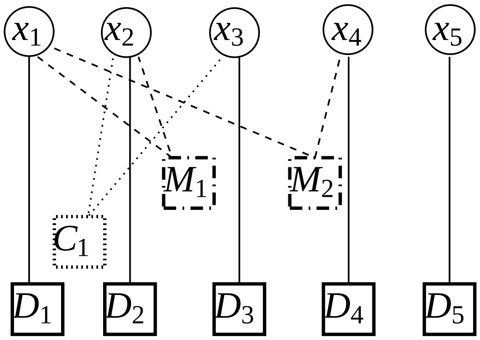

Figure A.1: Factor graph for the HMRF in Figure 4.1

the constraint) to potential values. For the factor node encoding the must-link constraint between x_i and x_j, the potential value for the entry (y_i,y_j) in the constraint potential table is 1 if $y_i = y_j$, i.e., x_i and x_j have the same cluster assignments. If $y_i \neq y_j$, the potential value is $\exp(-w_{ij})$, where w_{ij} is the weight of the constraint. Similarly, for the cannot-link factor nodes, the potential tables have values of 1 for the entry (y_i,y_j) where $y_i \neq y_j$, and $\exp(-w_{ij})$ if $y_i = y_j$.

Finding the collective assignment of points to minimize $\mathcal{J}_{\text{hmrf-kmeans}}$ in the E-step corresponds to running the max-product message-passing algorithm on the factor graph (Kschischang et al., 2001). Once the message-passing algorithm converges, the cluster assignment for each data point is obtained from the value in the corresponding variable node.

A.2 Linear programming relaxation approach

The task of finding an assignment of instances to clusters to minimize the objective function can be posed as an integer programming problem. Such a formulation has been proposed by Kleinberg and Tardos in the context of the general *metric labeling* problem, where they considered the cost of assigning labels to instances while attempting to satisfy a set of must-link pairwise constraints (Kleinberg & Tardos, 1999). We extend this formulation to include cannot-link constraints, which allows using it for assigning instances to clusters in the E-step of HMRF-KMEANS.

Let $U = \{u_{ih}\}$, $i = 1,\ldots,n$, $h = 1,\ldots,k$, be a set of nonnegative binary variables that encode membership of instances in clusters: $u_{ih} = 1$ signifies that the i^{th} instance belongs to the h^{th} cluster. Sets of nonnegative binary variables $U^{(M)} = \{u_i^{(M)}\}_{i=1}^{|C_{ML}|}$ and $U^{(C)} = \{u_i^{(C)}\}_{i=1}^{|C_{CL}|}$ encode violations of must-link and cannot-link pairwise constraints respectively. Each $u_k^{(M)} = 1$ signifies that the k^{th} must-link pairwise constraint $e_k = (x_{k_1}, x_{k_2})$ is violated, while $u_k^{(C)} = 1$ signifies that the k^{th} cannot-link pairwise constraint $e_k = (x_{k_1}, x_{k_2})$ is violated. The objective function to be optimized in the E-step of HMRF-KMEANS then becomes:

$$\mathcal{J}_{\text{hmrf-kmeans}} = \sum_{x_i \in X} \sum_{h \in L} D(x_i, \mu_h)\ u_{ih} + \sum_{(x_{k_1}, x_{k_2}) \in C_{ML}} w_k u_k^{(M)} + \sum_{(x_{k_1}, x_{k_2}) \in C_{CL}} w_k u_k^{(C)}, \qquad \text{(A.1)}$$

where $L = \{1,\ldots,k\}$. Assigning each instance to only one cluster imposes the following linear constraint on variables in U:

$$\sum_{h \in L} u_{ih} = 1, \quad x_i \in X. \qquad \text{(A.2)}$$

Also, consistency of pairwise constraint violation variables in $U^{(M)}$ and $U^{(C)}$ with the assignment variables in U requires satisfaction of the following linear constraints:

$$\begin{aligned} u_k^{(M)} &= \frac{1}{2}\sum_{h\in L}|u_{k_1h}-u_{k_2h}|, e_k=(x_{k_1},x_{k_2})\in C_{ML}, \\ u_k^{(C)} &= 1-\frac{1}{2}\sum_{h\in L}|u_{k_1h}-u_{k_2h}|, e_k=(x_{k_1},x_{k_2})\in C_{CL}. \end{aligned} \tag{A.3}$$

These constraints can be expressed in a linear program by replacing variables $U^{(M)}$ and $U^{(C)}$ with corresponding sets of auxiliary variables $Z^{(M)}$ and $Z^{(C)}$, where $z_{kh}^{(M)}=1$ iff the k^{th} must-link pair $e_k=(x_{k_1},x_{k_2})$ is violated and either x_{k_1} or x_{k_2} is assigned to h^{th} cluster. Semantics of $z_{kh}^{(C)}$ are similar: $z_{kh}^{(C)}=1$ iff k^{th} cannot-link pair $e_k=(x_{k_1},x_{k_2})$ is violated and both x_{k_1} and x_{k_2} are assigned to h^{th} cluster. Variables in $U^{(M)}$ and $U^{(C)}$ can be expressed via variables in $Z^{(M)}$ and $Z^{(C)}$ as follows:

$$\begin{aligned} u_k^{(M)} &= \frac{1}{2}\sum_{h\in L} z_{kh}^{(M)}, \;\; e_k=(x_{k_1},x_{k_2})\in C_{ML}, \\ u_k^{(C)} &= \sum_{h\in L} z_{kh}^{(C)}, \qquad e_k=(x_{k_1},x_{k_2})\in C_{CL}. \end{aligned} \tag{A.4}$$

Consistency of assignment variables in U with pairwise constraint violation variables in $Z^{(M)}$ and $Z^{(C)}$ can then be achieved by introducing the following linear constraints:

$$z_{kh}^{(M)} \geq u_{k_1h} - u_{k_2h}, \qquad e_k = (x_{k_1}, x_{k_2}) \in C_{ML} \tag{A.5}$$

$$z_{kh}^{(M)} \geq u_{k_2h} - u_{k_1h}, \qquad e_k = (x_{k_1}, x_{k_2}) \in C_{ML} \tag{A.6}$$

$$z_{kh}^{(C)} \leq u_{k_1h} + u_{k_2h}, \qquad e_k = (x_{k_1}, x_{k_2}) \in C_{CL} \tag{A.7}$$

$$z_{kh}^{(C)} \geq u_{k_1h} + u_{k_2h} - 1, \qquad e_k = (x_{k_1}, x_{k_2}) \in C_{CL}. \tag{A.8}$$

Minimization of objective function Eqn. (A.1) under the constraints Eqn. (A.2) and Eqns. (A.5)-(A.8) to solve for binary variables U, $Z^{(M)}$, and $Z^{(C)}$ is NP-hard. Kleinberg and Tardos proposed a linear programming relaxation of this integer programming problem by allowing U, $Z^{(M)}$, and $Z^{(C)}$ to be non-negative real numbers, and provided a randomized method for rounding the real solution to the linear program to integers (Kleinberg & Tardos, 1999). We follow their approach, which allows us to perform collective assignment of all instances in X to cluster centroids.

Appendix B

Active learning for constraint acquisition

In this appendix, we provide some analysis of the 2-phase active learning algorithm presented in Chapter 5.

B.1 Model assumptions

First of all, we present the formal model of the dataset based on which the analysis of active learning will be done. The data is assumed to be coming from *k* disjoint uniform density balls of unequal size in a metric space. The balls are defined in terms of the metric. All data points inside any particular ball are assumed to be in the same cluster, and points from different balls are assumed to be from different clusters. The oracle is assumed to know this model. Let n be the total number of points under consideration. Let $\{\pi_h\}_{h=1}^{k}$ be the probabilities of drawing a point randomly from the h^{th} ball B_h. Without loss of generality, we assume $\pi_1 \leq \pi_2 \leq \cdots \leq \pi_k$. Further, let $1/l \leq \pi_1$. Let m_h be the number of points in the dataset from B_h. Then, $\pi_h = m_h/n$ and $\pi_h \propto V_h$, the volume of B_h, $\forall h$. Now, the number of possible *cannot-links* is $\sum_{\{h,l,h<l\}} m_h m_l$ and the number of *must-links* is $\sum_h \binom{m_h}{2}$. Let $\alpha = \sum_{\{h,l,h<l\}} m_h m_l / \sum_h \binom{m_h}{2}$.

B.2 Analysis of random initialization

We argue that within a small number of random queries, the probability of getting even a 3-point neighborhood from any cluster is very low. Given Q pairs at random, on average there will be one *must-link* in every $(1+\alpha)$ pairs. Hence, there will be a total of $Q/(1+\alpha)$ *must-link* pairs in the expected behavior. Then, for the h^{th} cluster, there will be $r_h = \pi_h Q/(1+\alpha) \ll m_h$ *must-link* pairs on average. We focus on a particular cluster B_h on which r_h pairs have been selected at random. The size of the cluster is $m_h = n/k$. We will not get a 3-point neighborhood from B_h if none of the points $x \in B_h$ gets drawn more than once in the random pair sampling. If the sampling of r_h pairs is replaced by the sampling of $2r_h$ vertices without replacement, the probability of getting a vertex twice is increased. Hence, the probability p_h of *not* getting a 3-point neighborhood is lower bounded by the probability of not getting a vertex twice in the vertex sampling setting. So,

$$\begin{aligned} p_h &\geq \sum_{\substack{\sum_l \beta_l = 2r_h \\ \beta_l < 2, \forall l}} \binom{2r_h}{\beta_1 \cdots \beta_{m_h}} \left(\frac{1}{m_h}\right)^{2r_h} \\ &= 1 \cdot \left(1 - \frac{1}{m_h}\right) \cdot \left(1 - \frac{2}{m_h}\right) \cdots \left(1 - \frac{2r_h - 1}{m_h}\right) \\ &\geq \left(1 - \frac{2r_h}{m_h}\right)^{2r_h} \approx 1 - \frac{4r_h^2}{m_h} = 1 - \frac{4m_h Q^2}{n^2(1+\alpha)^2} \end{aligned}$$

which is close to 1 for small values of Q. Hence, the probability of getting 3-point neighborhoods is very low. Therefore, the initialization is essentially done by k random draws from a set of approximately $Q/(1+\alpha)$ 2-point neighborhoods. In this setting, the probability of getting exactly one neighborhood from each cluster is

$$k! \prod_{h=1}^{k} \pi_h \leq \frac{k!}{k^k} = \frac{\sqrt{2\pi k}}{e^k}(1 + \frac{1}{12k} + O(\frac{1}{k^2}))$$

using the AM-GM inequality and the Stirling's formula. Clearly, the probability is quite low. This results in significant variance in the initializing neighborhoods and explains the initial jitter for the non-active algorithms for low values of Q.

B.3 Analysis of EXPLORE

Given 2 balls of unequal size, we will now try to see how many farthest-first traversals will be required to get atleast one point from each ball.

In the worst case, if the disjoint balls are placed by an adversary, the adversary will try to place the balls such that getting a point from at least one ball is very difficult. One can show that the optimum strategy for the adversary will be to make the smaller ball difficult to reach. Using a packing argument, we show that irrespective of the placement of the balls, the farthest first traversal cannot avoid any particular ball for long. Consider two balls b,B with probabilities π_b,π_B. Let r_b,r_B be the radii of the two balls, and V_b,V_B be the volumes of the two balls. Further, let $\sigma_b(B)$ denote the packing number of B with b balls — the maximum number of disjoint b balls that can be packed inside the ball B. Now, if there are just these two balls in the universe and if farthest-first traversal starts in B, the points obtained from B before entering b must have pairwise distances (between their centers) of at least $2r_b$, because otherwise the traversal would have picked the farthest point from b and got a distance of at least $2r_b$. Hence, the traversal cannot stay in B for more that $\sigma_b(B)$ farthest-first jumps because there are exactly these many points inside B that can be at a distance of at least $2r_b$ from each other. Now, the packing number $\sigma_b(B) \leq V_B/V_b = \pi_B/\pi_b$, the ratio of their probabilities. So, the farthest first traversal will atmost stay in the larger ball for atmost $\pi_B/\pi_b = (1-\pi_b)/\pi_b = 1/\pi_b - 1 \leq \ell - 1$ jumps before being forced to pick a point from the smaller ball. So, the farthest-first scheme is sure to get one point from each of the balls in at most ℓ traversals.

We are currently working on extending this argument to the general case of k balls.

Bibliography

Abe, N., & Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98)*, pp. 1–10.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press, New York.

Banerjee, A., Dhillon, I., Ghosh, J., & Sra, S. (2003). Generative model-based clustering of directional data. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pp. 19–28 Washington, DC.

Banerjee, A. (2001). Large scale clustering of sequential patterns. Doctoral Dissertation Proposal, University of Texas at Austin.

Banerjee, A., Krumpelman, C., Basu, S., Mooney, R. J., & Ghosh, J. (2005). Model-based overlapping clustering. Submitted for publication.

Banerjee, A., & Langford, J. (2004). An objective evaluation criterion for clustering. In

Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-04).

Banerjee, A., Merugu, S., Dhillon, I. S., & Ghosh, J. (2004). Clustering with Bregman divergences. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04)* Lake Buena Vista, FL.

Bansal, N., Blum, A., & Chawla, S. (2002). Correlation clustering. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science (FOCS-02)*, pp. 238–247.

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2003). Learning distance functions using equivalence relations. In *Proceedings of 20th International Conference on Machine Learning (ICML-2003)*, pp. 11–18.

Basu, S., Banerjee, A., & Mooney, R. J. (2002). Semi-supervised clustering by seeding. In *Proceedings of 19th International Conference on Machine Learning (ICML-2002)*, pp. 19–26.

Basu, S., Banerjee, A., & Mooney, R. J. (2004). Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM International Conference on Data Mining (SDM-04)*.

Basu, S., Bilenko, M., & Mooney, R. J. (2003). Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering. In *Proceedings of the*

ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining, pp. 42–49 Washington, DC.

Basu, S., Bilenko, M., & Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, pp. 59–68 Seattle, WA.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B (Methodological)*, *48*(3), 259–302.

Bilenko, M., Basu, S., & Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of 21st International Conference on Machine Learning (ICML-2004)*, pp. 81–88 Banff, Canada.

Bilenko, M., & Mooney, R. J. (2002). Learning to combine trained distance metrics for duplicate detection in databases. Tech. rep. AI 02-296, Artificial Intelligence Laboratory, University of Texas at Austin, Austin, TX.

Bilenko, M., & Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, pp. 39–48 Washington, DC.

Bilmes, J. (1997). A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Tech. rep. ICSI-TR-97-021, ICSI.

Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/˜mlearn/MLRepository.html.

Blum, A., Lafferty, J., Rwebangira, M., & Reddy, R. (2004). Semi-supervised learning using randomized mincuts. In *Proceedings of 21st International Conference on Machine Learning (ICML-2004)*.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pp. 92–100 Madison, WI.

Boley, D. (1998). Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, *2*(4), 325–344.

Boykov, Y., Veksler, O., & Zabih, R. (1998). Markov random fields with efficient approximations. In *Proceedings of IEEE Computer Vision and Pattern Recognition Conference (CVPR-98)*, pp. 648–655.

Bradley, P. S., Mangasarian, O. L., & Street, W. N. (1997). Clustering via concave minimization. In Mozer, M. C., Jordan, M. I., & Petsche, T. (Eds.), *Advances in Neural Information Processing Systems 9*, pp. 368–374. The MIT Press.

Buntine, W. L. (1994). Operations for learning graphical models. *Journal of Artificial Intelligence Research*, *2*, 159–225.

Chan, P., Schlag, M., & Zien, J. (1994). Spectral *k*-way ratio cut partitioning. *IEEE Transactions on CAD-Integrated Circuits and Systems*, *13*, 1088–1096.

Chang, H., & Yeung, D.-Y. (2004). Locally linear metric adaptation for semi-supervised clustering. In *Proceedings of 21st International Conference on Machine Learning (ICML-2004)*.

Charikar, M., Guruswami, V., & Wirth, A. (2003). Clustering with qualitative information. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS-03)*, pp. 524–533.

Cohn, D., Caruana, R., & McCallum, A. (2003). Semi-supervised clustering with user feedback. Tech. rep. TR2003-1892, Cornell University.

Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145.

Cover, T. M., & Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience.

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey, J. W. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of Fifteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 318–329.

Dasgupta, S. (2002). Performance guarantees for hierarchical clustering. In *Proceedings of the Annual Conference on Computational Learning Theory (COLT)*, pp. 351–363.

Davidson, I., & Ravi, S. (2005). Clustering with constraints: Feasibility issues and the k-means algorithm. In *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM-05)*.

Demiriz, A., Bennett, K. P., & Embrechts, M. J. (1999). Semi-supervised clustering using genetic algorithms. In *Artificial Neural Networks in Engineering (ANNIE-99)*, pp. 809–814.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, *39*, 1–38.

Devroye, L., Gyorfi, L., & Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer Verlag.

Dhillon, I. S., Fan, J., & Guan, Y. (2001). Efficient clustering of very large document collections. In *Data Mining for Scientific and Engineering Applications*. Kluwer Academic Publishers.

Dhillon, I. S., & Modha, D. S. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning*, *42*, 143–175.

Dhillon, I. S., & Guan, Y. (2003). Information theoretic clustering of sparse co-occurrence data. In *Proceedings of the Third IEEE International Conference on Data Mining (ICDM-2003)*, pp. 517–521.

Dom, B. E. (2001). An information-theoretic external cluster-validity measure. Research report RJ 10219, IBM.

Dubnov, S., El-Yaniv, R., Gdalyahu, Y., Schneidman, E., Tishby, N., & Yona, G. (2002). A new nonparametric pairwise clustering algorithm based on iterative estimation of distance profiles. *Machine Learning*, *47*(1), 35–61.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification* (Second edition). Wiley, New York.

Fayyad, U. M., Reina, C., & Bradley, P. S. (1998). Initialization of iterative refinement clustering algorithms. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pp. 194–198.

Fern, X., & Brodley, C. (2003). Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of 20th International Conference on Machine Learning (ICML-2003)*.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, *2*, 139–172.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, *28*, 133–168.

Garey, M. R., Johnson, D. S., & Witsenhausen, H. S. (1982). The complexity of the generalized Lloyd-max problem. *IEEE Transactions on Information Theory*, *28*(2), 255–256.

Garey, M., & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, NY.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–742.

Ghahramani, Z., & Jordan, M. I. (1994). Supervised learning from incomplete data via the EM approach. In *Advances in Neural Information Processing Systems 6*, pp. 120–127.

Ghani, R., Jones, R., & Rosenberg, C. (Eds.). (2003). *ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, Washington, DC.

Goldszmidt, M., & Sahami, M. (1998). A probabilistic approach to full-text document clustering. Tech. rep. ITAD-433-MS-98-044, SRI International.

Gondek, D., Vaithyanathan, S., & Garg, A. (2005). Clustering with model-level constraints. In *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM-05)*.

Guha, S., Rastogi, R., & Shim, K. (1999). Rock: a robust clsutering algorithm for categorical attributes. In *Proceedings of the Fifteenth International Conference on Data Engineering*.

Hammersley, J., & Clifford, P. (1971). Markov fields on graphs and lattices. Unpublished manuscript.

Hastie, T., & Tibshirani, R. (1996). Discriminant adaptive nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(6), 607–617.

Hertz, T., Bar-Hillel, A., & Weinshall, D. (2004). Boosting margin based distance functions for clustering. In *Proceedings of 21st International Conference on Machine Learning (ICML-2004)*.

Hinneburg, A., & Keim, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pp. 58–65.

Hiu, M., Law, C., Topchy, A., & Jain, A. K. (2005). Model-based clustering with probabilistic constraints. In *Proceedings of the 2005 SIAM International Conference on Data Mining (SDM-05)*.

Hochbaum, D., & Shmoys, D. (1985). A best possible heuristic for the *k*-center problem. *Mathematics of Operations Research*, *10(2)*, 180–184.

Hofmann, T., & Buhmann, J. M. (1998). Active data clustering. In *Advances in Neural Information Processing Systems 10*.

Jaakkola, T., & Haussler, D. (1999). Exploiting generative models in discriminative classifiers. In *Advances in Neural Information Processing Systems 11*, pp. 487–493.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall, New Jersey.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, *31*(3), 264–323.

Jain, K., & Vazirani, V. (2001). Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *Journal of the ACM*, *48*, 274–296.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)*, pp. 200–209 Bled, Slovenia.

Kamvar, S. D., Klein, D., & Manning, C. D. (2002). Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. In *Proceedings of 19th International Conference on Machine Learning (ICML-2002)*.

Kamvar, S. D., Klein, D., & Manning, C. D. (2003). Spectral learning. In *Proceedings* *of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-2003)*, pp. 561–566 Acapulco, Mexico.

Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Computer*, *32*(8), 68–75.

Karypis, G., & Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, *20*(1), 359–392.

Kaufman, L., & Rousseeuw, P. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York.

Kearns, M., Mansour, Y., & Ng, A. Y. (1997). An information-theoretic analysis of hard and soft assignment methods for clustering. In *Proceedings of 13th Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pp. 282–293.

Klein, D., Kamvar, S. D., & Manning, C. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the The Nineteenth International Conference on Machine Learning (ICML-2002)*, pp. 307–314 Sydney, Australia.

Kleinberg, J., & Tardos, E. (1999). Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS-99)*, pp. 14–23.

Kschischang, F. R., Frey, B., & Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, *47*(2), 498–519.

Kulis, B., Basu, S., Dhillon, I., & Mooney, R. J. (2005). Semi-supervised graph clustering: A kernel approach. Proceedings of Twenty-Second International Conference on Machine Learning (ICML-05).

Kumar, S., & Hebert, M. (2003). Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Vol. 2, pp. 1150–1157.

Lewis, D., & Gale, W. (1994). A sequential algorithm for training text classifiers. In *Proceedings of Seventeenth International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-94)*.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297.

Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

McCallum, A., & Nigam, K. (1998). Employing EM and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98)* Madison, WI. Morgan Kaufmann.

Meila, M. (2003). Comparing clusterings by the variation of information. In *Proceedings of the 16th Annual Conference on Computational Learning Theory*, pp. 173–187.

Meila, M., & Heckerman, D. (1998). An experimental comparison of several clustering and initialization methods. Tech. rep. MSR-TR-98-06, Microsoft Research.

Mettu, R., & Plaxton, C. G. (2000). The online median problem. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pp. 339–348.

Miller, D. J., & Browning, J. (2003). A mixture model and EM algorithm for robust classification, outlier rejection, and class discovery. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Mirchandani, P., & Francis, R. (Eds.). (1990). *Discrete Location Theory*. Wiley, New York, NY.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill, New York, NY.

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 13*, Vol. 14.

Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems 14*, pp. 605–610.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*, 103–134.

Nigam, K. (2001). *Using Unlabeled Data to Improve Text Classification*. Ph.D. thesis, Carnegie Mellon University.

P. Cheeseman, J. Kelly, M. S. J. S. W. T., & Freeman, D. (1988). Autoclass: A Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning (ICML-88)*, pp. 54–56.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo,CA.

Pelleg, D., & Moore, A. (2000). X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000)*.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Salton, G., & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw Hill, New York.

Seeger, M. (2000). Learning with labeled and unlabeled data. Tech. rep., Institute for ANC, Edinburgh, UK. See `http://www.dai.ed.ac.uk/˜seeger/papers.html`.

Segal, E., Wang, H., & Koller, D. (2003a). Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, *19*, i264–i272.

Segal, E., Battle, A., & Koller, D. (2003b). Decomposing gene expression into cellular processes. In *Proceedings of the 8th Pacific Symposium on Biocomputing*.

Selim, S., & Ismail, M. (1984). K-means-type algorithms: A generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 81–87.

Sheikholesami, G., Chatterjee, S., & Zhang, A. (1998). Wavecluster: A muti-resolution clustering approach for very large spatial databases. In *Proceedings of the International Conference on Very Large Data Bases*, pp. 428–439.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Sinkkonen, J., & Kaski, S. (2000). Semisupervised clustering based on conditional distributions in an auxiliary space. Tech. rep. A60, Helsinki University of Technology.

Strehl, A., & Ghosh, J. (2000). A scalable approach to balanced, high-dimensional clustering of market-baskets. In *Proceedings of the Seventh International Conference on High Performance Computing (HiPC 2000)*.

Strehl, A., Ghosh, J., & Mooney, R. (2000). Impact of similarity measures on web-page clustering. In *Workshop on Artificial Intelligence for Web Search (AAAI 2000)*, pp. 58–64.

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons.

Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001). Constrained K-Means clustering with background knowledge. In *Proceedings of 18th International Conference on Machine Learning (ICML-2001)*, pp. 577–584.

Wainwright, M. J., & Jordan, M. I. (2003). Graphical models, exponential families, and variational inference. Tech. rep. 649, Department of Statistics, University of California, Berkeley.

Wallace, C. S., & Lowe, D. L. (1999). Minimum message length and Kolmogorov complexity. In *The Computer Journal*. Oxford University Press.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2003). Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pp. 505–512 Cambridge, MA. MIT Press.

Xu, L., Neufeld, J., Larson, B., & Schuurmans, D. (2005). Maximum margin clustering. In *Advances in Neural Information Processing Systems 17*, pp. 1537–1544.

Yan, R., Zhang, J., Yang, J., & Hauptmann, A. G. (2004). A discriminative learning framework with pairwise constraints for video object classification. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, T., Ramakrishnan, R., & Livny, M. (1996). Birch: An efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 103–114.

Zhang, Y., Brady, M., & Smith, S. (2001). Hidden Markov random field model and segmentation of brain MR images. *IEEE Transactions on Medical Imaging*, *20*(1), 45–57.

Vita

Sugato Basu is a PhD student in the Computer Science Department of University of Texas (UT) at Austin. His PhD advisor is Prof. Raymond J. Mooney and his research interests are in the area of Machine Learning and Data Mining. Before this, he received his MS from the Computer Engineering department of University of California at Santa Cruz, and his BTech (Honors) in Computer Science and Engineering from the Indian Institute of Technology at Kharagpur. He received the Best Research Paper Award at KDD 2004, the IBM PhD Fellowship in 2002 and the MCD Fellowship from the University of Texas in 2000. He is on the Program Committee of KDD and AAAI in 2005 and is the reviewer for multiple journals, including the Journal of Machine Learning Research, Pattern Recognition Letters, Journal of Data Mining and Knowledge Discovery, and IEEE Pattern Analysis and Machine Intelligence. 2 patents have been filed based on his summer research at Google.

Permanent Address: P-125 Lake Terrace,
Calcutta 700029,
West Bengal, India.

This dissertation was typeset with $\LaTeX\,2_\varepsilon$[1] by the author.

[1] $\LaTeX\,2_\varepsilon$ is an extension of $\LaTeX$. $\LaTeX$ is a collection of macros for $\TeX$. $\TeX$ is a trademark of the American Mathematical Society. The macros used in formatting this dissertation were written by Dinesh Das, Department of Computer Sciences, The University of Texas at Austin, and extended by Bert Kay and James A. Bednar.