In Semi-Supervised Learning, O. Chapelle, B. Schoelkopf, and A. Zien (eds.), MIT Press, 2006, to appear.

3 Probabilistic Semi-Supervised Clustering with Constraints

Sugato Basu
Mikhail Bilenko
Arindam Banerjee
Raymond Mooney

In certain clustering tasks it is possible to obtain limited supervision in the form of pairwise constraints, i.e., pairs of instances labeled as belonging to same or different clusters. The resulting problem is known as semi-supervised clustering, *an instance of semi-supervised learning stemming from a traditional unsupervised learning setting. Several algorithms exist for enhancing clustering quality by using supervision in the form of constraints. These algorithms typically utilize the pairwise constraints to either modify the clustering objective function or to learn the clustering distortion measure. This chapter describes an approach that employs Hidden Markov Random Fields (HMRFs) as a probabilistic generative model for semi-supervised clustering, thereby providing a principled framework for incorporating constraint-based supervision into prototype-based clustering. The HMRF-based model allows the use of a broad range of clustering distortion measures, including Bregman divergences (e.g., squared Euclidean distance, KL divergence) and directional distance measures (e.g., cosine distance), making it applicable to a number of domains. The model leads to the* HMRF-KMeans *algorithm which minimizes an objective function derived from the joint probability of the model, and allows unification of constraint-based and distance-based semi-supervised clustering methods. Additionally, a two-phase active learning algorithm for selecting informative pairwise constraints in a query-driven framework is derived from the HMRF model, facilitating improved clustering performance with relatively small amounts of supervision from the user.*

3.1 Introduction

semi-supervised clustering with constraints

This chapter focuses on *semi-supervised clustering with constraints*, the problem of partitioning a set of data points into a specified number of clusters when limited supervision is provided in the form of pairwise constraints. While clustering is traditionally considered to be a form of unsupervised learning since no class labels are given, inclusion of pairwise constraints makes it a semi-supervised learning task, where the performance of unsupervised clustering algorithms can be improved using the limited training data.

must-link and cannot-link constraints

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. Alternatively, must-link and cannot-link constraints are sometimes called *equivalence* and *non-equivalence* constraints respectively. Typically, the constraints are "soft", that is, clusterings that violate them are undesirable but not prohibited.

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 dataset.

constraint-based and distance-based methods

Proposed methods for semi-supervised clustering fall into two general categories that we call *constraint-based* and *distance-based*. Constraint-based methods use the provided supervision to guide the algorithm towards a data partitioning that avoids violating the constraints [Demiriz et al., 1999, Wagstaff et al., 2001, Basu et al., 2002]. In distance-based approaches, an existing clustering algorithm that uses a particular distance function between points is employed; however, the distance function is parameterized and the parameter values are learned to bring must-linked points together and take cannot-linked points further apart [Bilenko and Mooney, 2003, Cohn et al., 2003, Klein et al., 2002, Xing et al., 2003].

This chapter describes an approach to semi-supervised clustering based on Hidden Markov Random Fields (HMRFs) that combines the constraint-based and distance-based approaches in a unified probabilistic model. The probabilistic formulation leads to a clustering objective function derived from the joint probability of observed data points, their cluster assignments, and generative model parameters. This objective function can be optimized using an EM-style clustering algorithm, HMRF-KMeans, that finds a local minimum of the objective function. HMRF-KMeans can be used to perform semi-supervised clustering using a broad class of distortion (distance) functions,[1] namely *Bregman divergences* [Banerjee et al., 2005b], which include a wide variety of useful distances, e.g., KL divergence, squared Euclidean distance, I-divergence, and Itakuro-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 and Ribeiro-Neto, 1999]. Clustering algorithms have been developed that utilize distortion measures appropriate for directional data [Dhillon and Modha, 2001, Banerjee et al., 2005a], and the HMRF-KMeans framework naturally extends them.

A practical aspect of semi-supervised clustering with constraints is how maximally informative constraints can be acquired in a real-life setting, where a limited set of queries can be made to a user in an interactive learning setting [McCallum and Nigam, 1998]. In that case, fewer queries should be posed to the user to obtain constraints that can significantly enhance the clustering accuracy. To this end, a new method for active learning is presented—it selects good pairwise constraints for semi-supervised clustering by asking queries to the user of the form "Are these two examples in same or different classes?" leading to improved clustering performance.

3.2 HMRF Model for Semi-supervised Clustering

Partitional prototype-based clustering is the underlying unsupervised clustering setting under consideration. In such a setting, a set of data points is partitioned into a pre-specified number of clusters, where each cluster has a representative (or "prototype"), so that a well-defined cost function, involving a distortion measure between the points and the cluster representatives, is minimized. A well-known unsupervised clustering algorithm that follows this framework is K-Means [MacQueen, 1967].

problem setting

Our semi-supervised clustering model considers a sample of n data points $X = (x_1, \ldots, x_n)$, 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_A used to compute distance between points: $d_A : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, where A is the set of distortion measure parameters. 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

1. In this chapter, "distance measure" is used synonymously with "distortion measure": both terms refer to the distance function used for clustering.

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 datapoints X into K disjoint clusters $(X_1, \ldots, X_K)$ so that the total distortion between the points and the corresponding cluster representatives is minimized according to the given distortion measure d_A, while constraint violations are kept to a minimum.

3.2.1 HMRF Model Components

The Hidden Markov Random Field (HMRF) probabilistic framework [Zhang et al., 2001] for semi-supervised constrained clustering consists of the following components:

- An *observable* set $X = (x_1, \ldots, x_n)$ 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 $Y = (y_1, \ldots, y_n)$ 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 $(1, \ldots, K)$.
- An *unobservable* (hidden) set of generative model parameters Θ, which consists of distortion measure parameters A and cluster representatives $M = (\mu_1, \ldots, \mu_K)$: $\Theta = \{A, M\}$.
- 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.

HMRF example

Fig. 3.1 shows a simple example of an HMRF. X consists of five datapoints 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. 3.1 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.

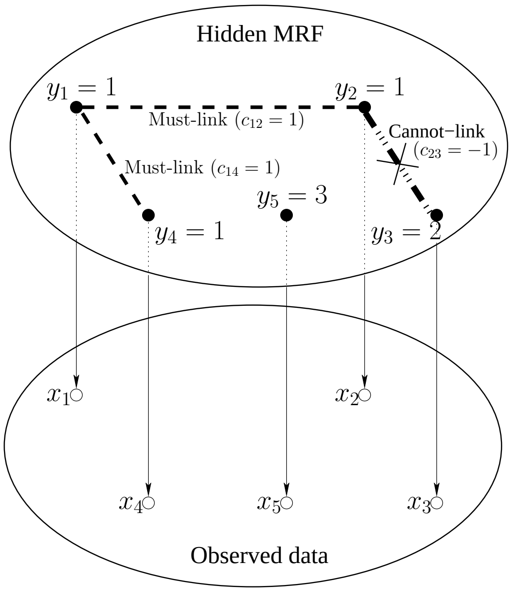

Figure 3.1 A Hidden Markov Random Field

3.2.2 Markov Random Field over Labels

Markov 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), where the conditional probability distribution over the hidden variables obeys the Markov property:

$$\forall i,\ \mathrm{P}(y_i | Y - \{y_i\}, \Theta, C) = \mathrm{P}(y_i | N_i, \Theta, C). \tag{3.1}$$

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 and Clifford, 1971], the prior probability of a particular label configuration Y can be expressed as a Gibbs distribution [Geman and Geman, 1984], so that

$$\mathrm{P}(Y | \Theta, C) = \frac{1}{Z} \exp\left(-v(Y)\right) = \frac{1}{Z} \exp\left(-\sum_{N_i \in N} v_{N_i}(Y)\right), \tag{3.2}$$

where N is the set of all neighborhoods, Z is the partition function (normalizing term), and $v(Y)$ is the overall label configuration potential function, which can be decomposed into a sum of functions $v_{N_i}(Y)$, each denoting the potential for

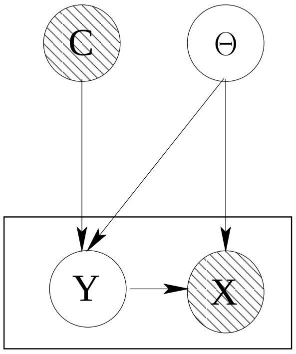

Figure 3.2 Graphical plate model of variable dependence

every neighborhood N_i in the label configuration Y. Since the potentials for every neighborhood are based on pairwise constraints in C (and model parameters Θ), the label configuration can be further decomposed as:

$$\mathrm{P}(Y|\Theta, C) = \frac{1}{Z} \exp\left(-\sum_{i,j} v(i,j)\right), \tag{3.3}$$

constraint potential function

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

$$v(i,j) = \begin{cases} w_{ij} f_{ML}(i,j) & \text{if } c_{ij} = 1 \text{ and } y_i \neq y_j \\ w_{ij} f_{CL}(i,j) & \text{if } c_{ij} = -1 \text{ and } y_i = y_j \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

The penalty functions f_{ML} and f_{CL} encode the lowered probability of observing configurations of Y where constraints encoded by C are violated. To this end, function f_{ML} penalizes violated must-link constraints and function f_{CL} penalizes violated cannot-link constraints. These functions are chosen to correspond with the distortion measure by employing same model parameters Θ, and will be described in detail in Section 3.3. 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.

3.2.3 Joint Probability in HMRF

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

$$\mathrm{P}(X, Y, \Theta|C) = \mathrm{P}(\Theta|C)\ \mathrm{P}(Y|\Theta, C)\ \mathrm{P}(X|Y, \Theta, C) \tag{3.5}$$

graphical plate model

The graphical plate model [Buntine, 1994] of the dependence between the random variables in the HMRF is shown in Figure 3.2, where the unshaded 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 probability of Θ is assumed to be independent of C. The probability of observing the label configuration Y depends on the constraints C and current generative model parameters Θ. Observed datapoints corresponding to variables X are generated using the model parameters Θ based on cluster labels Y, independent of the constraints C. The variables X are assumed to be mutually independent: each x_i is generated individually from a conditional probability distribution $\mathrm{P}(x|y,\Theta)$. Then, the conditional probability $\mathrm{P}(X|Y,\Theta,C)$ can be written as:

$$\mathrm{P}(X|Y,\Theta,C) = \mathrm{P}(X|Y,\Theta) = \prod_{i=1}^{n} p(x_i|y_i,\Theta), \tag{3.6}$$

where $p(\cdot|y_i,\Theta)$ is the parameterized 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_A, as described below in Section 3.2.4.

From (3.3), (3.5), and (3.6), it follows that maximizing the joint probability on the HMRF is equivalent to maximizing:

$$\mathrm{P}(X,Y,\Theta|C) = \mathrm{P}(\Theta)\left(\frac{1}{Z}\exp\left(-\sum_{c_{ij}\in C} v(i,j)\right)\right)\left(\prod_{i=1}^{n} p(x_i|y_i,\Theta)\right) \tag{3.7}$$

joint probability factorization

The joint probability in (3.7) has 3 factors. The first factor describes a probability distribution over the model parameters preventing them from converging to degenerate values, thereby providing regularization. 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 (3.7) is equivalent to jointly maximizing the likehood of generating datapoints from the model and the probability of label assignments that respect the constraints, while regularizing the model parameters.

3.2.4 Semi-supervised Clustering Objective Function on HMRF

Formulation (3.7) suggests a general framework for incorporating constraints into clustering. The choice of the conditional probability $p(x|y,\Theta)$ in a particular instantiation of the framework is directly connected to the choice of the distortion measure appropriate for the clustering task.

generative probability for X

When considering the conditional probability $p(x_i|y_i,\Theta)$—the probability of generating a datapoint x_i from the y_i-th cluster—our attention is restricted to probability densities from the exponential family, where the expectation parameter corresponding to the y_i-th cluster is μ_{y_i}, the mean of the points of that cluster. Using this assumption and the bijection between regular exponential distributions and regular Bregman divergences [Banerjee et al., 2005b], the conditional density for observed data can be represented as:

$$p(x_i|y_i,\Theta) = \frac{1}{Z_\Theta} \exp\bigl(-d_A(x_i,\mu_h)\bigr), \tag{3.8}$$

where $d_A(x_i,\mu_{y_i})$ is the Bregman divergence between x_i and μ_{y_i}, corresponding to the exponential density p, and Z_Θ is the normalizer.[2] Different clustering models fall into this exponential form:

- If x_i and μ_{y_i} are vectors in Euclidean space, and d_A is the square of the L_2 distance parameterized by a positive semidefinite weight matrix A ($d_A(x_i,\mu_{y_i}) = \|x_i - \mu_{y_i}\|_A^2$), then the cluster conditional probability is a Gaussian with covariance encoded by A^{-1} [Kearns et al., 1997];
- If x_i and μ_{y_i} are probability distributions and d_A is the KL-divergence ($d_A(x_i,\mu_{y_i}) = \sum_{m=1}^d x_{im} \log \frac{x_{im}}{\mu_{y_i m}}$), then the cluster conditional probability is a multinomial distribution [Dhillon and Guan, 2003].

The relation in (3.8) holds even if d_A is not a Bregman divergence but a directional distance measure like cosine distance. For example, if x_i and μ_{y_i} are vectors of unit length and d_A is one minus the dot-product of the vectors $\bigl(d_A(x_i,\mu_{y_i}) = 1 - \frac{\sum_{m=1}^d x_{im}\mu_{y_i m}}{\|x_i\|\|\mu_{y_i}\|}\bigr)$, then the cluster conditional probability is a von-Mises Fisher (vMF) distribution with unit concentration parameter [Banerjee et al., 2005a], which is essentially the spherical analog of a Gaussian. The connection between specific distortion measures studied in this paper and their corresponding cluster conditional probabilities is discussed in more detail in Section 3.3.3.

Putting (3.8) into (3.7) and taking logarithms gives the following cluster objective function, minimizing which is equivalent to maximizing the joint probability over the HMRF in (3.7):

$$\mathcal{J}_{\text{obj}} \quad = \quad \sum_{x_i \in X} d_A(x_i,\mu_{y_i}) + \sum_{c_{ij}\in C} v(i,j) - \log \mathrm{P}(\Theta) + \log Z + n \log Z_\Theta \tag{3.9}$$

Thus, the task is to minimize $\mathcal{J}_{\text{obj}}$ over the hidden variables Y and Θ (note that given Y, the means $M = (\mu_1,\ldots,\mu_K)$ are uniquely determined).

2. When $A = I$ (identity matrix), the bijection result [Banerjee et al., 2005b] ensures that the normalizer Z_Θ is 1. In general, there are additional multiplicative terms that depend only on x, and hence can be safely ignored for parameter estimation purposes.

3.3 HMRF-KMeans Algorithm

Since the cluster assignments and the generative model parameters are unknown in a clustering setting, minimizing (3.9) is an "incomplete-data problem". A popular solution technique for such problems the is *Expectation Maximization* (EM) algorithm [Dempster et al., 1977]. The K-Means 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., 2005b]. This section describes a K-Means-type hard partitional clustering algorithm, HMRF-KMeans, that finds a local minimum of the semi-supervised clustering objective function $\mathcal{J}_{\text{obj}}$ in (3.9).

3.3.1 Normalizing Component Estimation

Before describing the details of the clustering algorithm, it is important to consider the normalizing components: the MRF partition function $\log Z$ and the distortion function normalizer $\log Z_\Theta$ in (3.9). Estimation of the partition function cannot be performed in closed form for most non-trivial dependency structures, and approximate inference methods must be employed for computing it [Wainwright and Jordan, 2003].

normalizer approximation

Estimation of the distortion normalizer $\log Z_\Theta$ depends on the distortion measure d_A used by the model. This chapter considers three parameterized distortion measures: parameterized squared Euclidean distance, parameterized cosine distance, and parameterized 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 $\mathcal{J}_{\text{obj}}$ in (3.9). For the other distortion measures, estimating the distortion normalizer Z_Θ cannot be performed in closed form, and approximate inference must be again used [Banerjee et al., 2005a].

Since approximate inference methods can be very expensive computationally, two simplifying assumptions can be made: the MRF partition function 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 $\mathcal{J}_{\text{obj}}$ in (3.9) no longer exactly corresponds to a joint probability on a HMRF. However, minimizing this simplified objective has been shown to work well empirically [Bilenko et al., 2004, Basu et al., 2004b]. 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 approximate inference methods.

3.3.2 Parameter Priors

Following the definition of Θ in Section 3.2.1, the prior term $\log \mathrm{P}(\Theta)$ in (3.9) and the subsequent equations can be factored as follows:

$$\log \mathrm{P}(\Theta) = \log\big(\mathrm{P}(A)\mathrm{P}(M)\big) = \log \mathrm{P}(A) + P_M$$

where the distortion parameters A are assumed to be independant of the cluster centroids $M = (\mu_1, \ldots, \mu_K)$, and uniform priors are considered over the cluster centroids (leading to the constant term P_M). For different distortion measures, parameter values may exist that lead to degenerate solutions of the optimization problem. For example, for squared Euclidean distance, the zero matrix $A = \mathbf{0}$ is one such solution. To prevent degenerate solutions, $\mathrm{P}(A)$ is used to regularize the parameter values using a prior distribution.

Rayleigh prior

If the standard Gaussian prior was used on the parameters of the distortion function, it would allow the parameters to take negative values. Since it is desirable to constrain the parameter values to be non-negative, it is more appropriate to use the Rayleigh distribution [Papoulis and Pillai, 2001]. Assuming independence of the parameters $a_{ij} \in A$, the prior term based on the Rayleigh distribution is the following:

$$\mathrm{P}(A) = \prod_{a_{ij} \in A} \frac{a_{ij} \exp\left(-\frac{a_{ij}^2}{s^2}\right)}{s^2} \tag{3.10}$$

where s is the width parameter.

3.3.3 Adaptive Distortion Measures

Selecting an appropriate distortion measure d_A for a clustering task typically involves knowledge about properties of the particular domain and dataset. For example, squared Euclidean distance is most appropriate for low-dimensional data with distribution close to Gaussian, while cosine distance best captures distance between data described by vectors in high-dimensional space where differences in angles are important but vector lengths are not.

distortion measure selection

Distortion measures from two families are considered in this chapter: *Bregman divergences* [Banerjee et al., 2005b], which include parameterized squared Euclidean distance and Kullback-Leibler divergence, and distortion measures based on *directional* similarity functions, which include cosine similarity and Pearson's correlation [Mardia and Jupp, 2000]. The distortion measure for directional functions is chosen to be the directional similarity measure subtracted from a constant sufficiently large so that the resulting value is non-negative. For both Bregman divergences and cosine distance, there exist efficient K-Means-type iterative relocation algorithms that minimize the corresponding clustering objective [Banerjee et al., 2005a,b], which the HMRF-KMeans naturally extends to incorporate pairwise supervision.

For many realistic datasets, off-the-shelf distortion measures may fail to capture the correct notion of similarity in a clustering setting. While some unsupervised measures like squared Euclidean distance and Pearson's distance attempt to correct distortion estimates using the global mean and variance of the dataset, these measures may still fail to estimate distances accurately if the attributes' true contributions to the distance is not correlated with their variance. Several semi-supervised clustering approaches exist that incorporate adaptive distortion measures, including parameterizations of Jensen-Shannon divergence [Cohn et al., 2003] and squared Euclidean distance [Bar-Hillel et al., 2003, Xing et al., 2003]. However, these techniques use only constraints to learn the distortion measure parameters and exclude unlabeled data from the parameter learning step, as well as separate the parameter learning step from the clustering process.

adaptive distortion measure

Going a step further, the HMRF model provides an integrated framework which incorporates *both* learning the distortion measure parameters and constraint-sensitive cluster assignments. In HMRF-KMeans, the parameters of the distortion measure are learned iteratively as the clustering progresses, utilizing both unlabeled data and pairwise constraints. The parameters are modified to decrease the parameterized distance between violated must-linked constraints and increase it between violated cannot-link constraints, while allowing constraint violations if they accompany a more cohesive clustering.

This section presents three examples of distortion functions and their parameterizations for use with HMRF-KMeans: squared Euclidean distance, cosine distance and KL divergence. Through parameterization, each of these functions becomes adaptive in a semi-supervised clustering setting, permitting clusters of varying shapes.

constraint potential function

Once a distortion measure is chosen for a given domain, the functions f_{ML} and f_{CL}, introduced in Section 3.2.2 for penalizing must-link and cannot-link constraint violations respectively, must be defined. These functions typically follow a functional form identical or similar to the corresponding distortion measure, and are chosen as follows:

$$f_{ML}(i,j) = \varphi(i,j) \tag{3.11}$$

$$f_{CL}(i,j) = \varphi^{\max} - \varphi(i,j) \tag{3.12}$$

where $\varphi : X \times X \to \mathbb{R}^+$ is a non-negative function that penalizes constraint violation, and $\varphi^{\max}$ is an upper bound on the maximum value of φ over any pair of points in the dataset; examples of such bounds for specific distortion functions are shown below. The function φ is chosen to correlate with the distortion measure, assigning higher penalties to violations of must-link constraints between points that are distant with respect to the current parameter values of the distortion measure. Conversely, penalties for violated cannot-link constraints are higher for points that have low distance between them. With this formulation of the penalty functions, constraint violations lead to changes in the distortion measure parameters that attempt to mend the violations. The φ function for different clustering distortion measures is discussed in the following sections.

Accordingly, the potential function $v(i,j)$ in (3.4) becomes:

$$v(i,j) = \begin{cases} w_{ij}\varphi(x_i,x_j) & \text{if } c_{ij}=1 \text{ and } y_i \neq y_j \\ w_{ij}\big(\varphi^{\max} - \varphi(x_i,x_j)\big) & \text{if } c_{ij}=-1 \text{ and } y_i = y_j \\ 0 & \text{otherwise} \end{cases}, \tag{3.13}$$

and the objective function for semi-supervised clustering in (3.9) can be expressed as:

$$\begin{aligned} \mathcal{J}_{\text{obj}} = & \sum_{x_i \in X} d_A(x_i, \mu(i)) + \sum_{\substack{(x_i,x_j)\in C_{ML} \\ s.t.\ y_i \neq y_j}} w_{ij}\varphi(x_i,x_j) \\ & + \sum_{\substack{(x_i,x_j)\in C_{CL} \\ s.t.\ y_i = y_j}} w_{ij}\big(\varphi^{\max} - \varphi(x_i,x_j)\big) - \log \mathrm{P}(A) + n\log Z_\Theta \end{aligned} \tag{3.14}$$

Note that as discussed in Section 3.3.1, the MRF partition function term $\log Z$ has been dropped from the objective function.

3.3.3.1 Parameterized Squared Euclidean Distance

Squared Euclidean distance is parameterized using a symmetric positive-definite matrix A as follows:

$$d_{euc_A}(x_i,x_j) = \|x_i - x_j\|_A^2 = (x_i - x_j)^T A (x_i - x_j). \tag{3.15}$$

This form of the parameterized squared Euclidean distance is equivalent to Mahalanobis distance with an arbitrary positive semidefinite weight matrix A in place of the inverse covariance matrix, and it was previously used for semi-supervised clustering by [Xing et al., 2003] and [Bar-Hillel et al., 2003]. Such formulation can also be viewed as a projection of every instance x onto a space spanned by $A^{1/2}$: $x \to A^{1/2}x$.

To use parameterized squared Euclidean distance as the adaptive distortion measure for clustering, the φ function that penalizes constraint violations is defined as $\varphi(x_i,x_j) = d_{euc_A}(x_i,x_j)$. One possible initialization of the upper bound for cannot-link penalties is $\varphi^{\max}_{euc_A} = \sum_{(x_i,x_j)\in C_{CL}} d_{euc_A}(x_i,x_j)$, which guarantees that the penalty is always positive. Using these definitions along with (3.14), the following objective function is obtained for semi-supervised clustering with adaptive squared Euclidean distance:

$$\begin{aligned}\mathcal{J}_{euc_A} = &\sum_{x_i \in X} d_{euc_A}(x_i, \mu(i)) + \sum_{\substack{(x_i,x_j)\in C_{ML} \\ s.t.\ y_i \neq y_j}} w_{ij} d_{euc_A}(x_i, x_j) \\ &+ \sum_{\substack{(x_i,x_j)\in C_{CL} \\ s.t.\ y_i = y_j}} w_{ij}\big(\varphi^{\max}_{euc_A} - d_{euc_A}(x_i, x_j)\big) - \log \mathrm{P}(A) - n \log \det(A)\end{aligned} \tag{3.16}$$

Note that as discussed in Section 3.3.1, the $\log Z_\Theta$ term is computable in closed-form for a Gaussian distribution with covariance matrix A^{-1}, which is the underlying cluster conditional probability distribution for parameterized squared Euclidean distance. The $\log \det(A)$ term (3.16) corresponds to the $\log Z_\Theta$ term in this case.

3.3.3.2 Parameterized Cosine Distance

Cosine distance can be parameterized using a symmetric positive-definite matrix A, which leads to the following distortion measure:

$$d_{\cos_A}(x_i, x_j) = 1 - \frac{x_i^T A x_j}{\|x_i\|_A \|x_j\|_A}. \tag{3.17}$$

Because for realistic high-dimensional domains computing the full matrix A would be computationally expensive, a diagonal matrix is considered in this case, such that $a = diag(A)$ is a vector of positive weights.

To use parameterized squared Euclidean distance as the adaptive distortion measure for clustering, the φ function is defined as $\varphi(x_i, x_j) = d_{\cos_A}(x_i, x_j)$. Using this definition along with (3.14), and setting $\varphi^{\max} = 1$ as an upper bound on $\varphi(x_i, x_j)$, the following objective function is obtained for semi-supervised clustering with adaptive cosine distance:

$$\begin{aligned}\mathcal{J}_{\cos_A} = &\sum_{x_i \in X} d_{\cos_A}(x_i, \mu(i)) + \sum_{\substack{(x_i,x_j)\in C_{ML} \\ s.t.\ y_i \neq y_j}} w_{ij} d_{\cos_A}(x_i, x_j) \\ &+ \sum_{\substack{(x_i,x_j)\in C_{CL} \\ s.t.\ y_i = y_j}} w_{ij}\big(1 - d_{\cos_A}(x_i, x_j)\big) - \log \mathrm{P}(A)\end{aligned} \tag{3.18}$$

Note that as discussed in Section 3.3.1, it is difficult to compute the $\log Z_\Theta$ term in closed-form for parameterized cosine distance. So, the simplifying assumption is made that $\log Z_\Theta$ is constant during the clustering process and the normalizer term is dropped from (3.18).

3.3.3.3 Parameterized KL-Divergence

In certain domains, data is described by probability distributions, e.g. text documents can be represented as probability distributions over words generated by a multinomial model [Pereira et al., 1993]. KL-divergence is a widely used distance measure for such data: $d_{KL}(x_i, x_j) = \sum_{m=1}^{d} x_{im} \log \frac{x_{im}}{x_{jm}}$, where x_i and x_j are probability distributions over d events: $\sum_{m=1}^{d} x_{im} = \sum_{m=1}^{d} x_{jm} = 1$. In previous work, Cohn et al. [2003] parameterized KL-divergence by multiplying the m-th component by a weight γ_m: $d'_{KL}(x_i, x_j) = \sum_{m=1}^{d} \gamma_m x_{im} \log \frac{x_{im}}{x_{jm}}$.

I-divergence

In our framework, KL distance is parameterized using a diagonal matrix A, where $a = diag(A)$ is a vector of positive weights. This parameterization of KL by A converts it to I-divergence, a function that also belongs to the class of Bregman divergences [Banerjee et al., 2005b]. I-divergence has the form: $d_I(x_i, x_j) = \sum_{m=1}^{d} x_{im} \log \frac{x_{im}}{x_{jm}} - \sum_{m=1}^{d} (x_{im} - x_{jm})$, where x_i and x_j no longer need to be probability distributions but can be any non-negative vectors.[3] The following parameterization of KL is used:

$$d_{I_A}(x_i, x_j) = \sum_{m=1}^{d} a_m x_{im} \log \frac{x_{im}}{x_{jm}} - \sum_{m=1}^{d} a_m (x_{im} - x_{jm}), \tag{3.19}$$

which can be interpreted as scaling every component of the original probability distribution by a weight contained in the corresponding component of A, and then taking I-divergence between the transformed distributions.

For every distortion measure, the clustering framework described in Section 3.2.4 requires defining an appropriate constraint potential function that is symmetric, since the constraint pairs are unordered. To meet this requirement, a sum of weighted I-divergences from x_i and x_j to the mean vector $\frac{x_i + x_j}{2}$ is used. This parameterized I-divergence to the mean, d_{IM_A}, is analogous to Jensen-Shannon divergence [Cover and Thomas, 1991], the symmetric KL-divergence to the mean, and is defined as follows:

$$d_{IM_A}(x_i, x_j) = \sum_{m=1}^{d} a_m \big(x_{im} \log \frac{2x_{im}}{x_{im} + x_{jm}} + x_{jm} \log \frac{2x_{jm}}{x_{im} + x_{jm}} \big). \tag{3.20}$$

To use parameterized squared Euclidean distance as the adaptive distortion measure for clustering, the φ function is defined as $\varphi(x_i, x_j) = d_{IM_A}(x_i, x_j)$. Using this definition along with (3.14), the following objective function is obtained for semi-supervised clustering with adaptive KL distance:

3. For probability distributions, I-divergence and KL-divergence are equivalent.

$$\mathcal{J}_{I_A} = \sum_{x_i \in X} d_{I_A}(x_i, \mu(i)) + \sum_{\substack{(x_i,x_j)\in C_{ML} \\ s.t.\ y_i \neq y_j}} w_{ij} d_{IM_A}(x_i, x_j)$$
$$+ \sum_{\substack{(x_i,x_j)\in C_{CL} \\ s.t.\ y_i = y_j}} w_{ij}\big(d_{IM_A}^{\max} - d_{IM_A}(x_i, x_j)\big) - \log \mathrm{P}(A) \tag{3.21}$$

The upper bound $d_{IM_A}^{\max}$ can be initialized as $d_{IM_A}^{\max} = \sum_{m=1}^{d} a_m$, which follows from the fact that unweighted Jensen-Shannon divergence is bounded above by 1 [Lin, 1991].

Note that as discussed in Section 3.3.1, it is difficult to compute the $\log Z_\Theta$ term in closed-form for parameterized KL distance. So, analogously to the parameterized cosine distance case, the simplifying assumption is made that $\log Z_\Theta$ is constant during the clustering process and that term is dropped from (3.21).

3.3.4 EM Framework

As discussed earlier in this section, $\mathcal{J}_{\text{obj}}$ can be minimized by a K-Means-type iterative algorithm HMRF-KMeans. The outline of the algorithm is presented in Fig. 3.3. The basic idea of HMRF-KMeans is as follows: the constraints are used to get a 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 $\mathcal{J}_{\text{obj}}$. In the M-step, the cluster representatives $M = (\mu_1, \ldots, \mu_K)$ are re-estimated from the cluster assignments to minimize $\mathcal{J}_{\text{obj}}$ for the current assignment. The clustering distortion measure d_A is subsequently updated in the M-step to reduce the objective function by modifying the parameters A of the distortion measure.

generalized EM

Note that this corresponds to the generalized EM algorithm [Neal and Hinton, 1998, Dempster et al., 1977], where the objective function is reduced but not necessarily minimized in the M-step. Effectively, the E-step minimizes $\mathcal{J}_{\text{obj}}$ over cluster assignments Y, the M-step (A) minimizes $\mathcal{J}_{\text{obj}}$ over cluster representatives M, and the M-step (B) reduces $\mathcal{J}_{\text{obj}}$ over the parameters A of the distortion measure d_A. 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.

3.3.5 Initialization

Good initial centroids are essential for the success of partitional clustering algorithms such as K-Means. Good centroids are inferred from both the constraints and unlabeled data during initialization. For this, a two stage initialization process is used.

Algorithm: HMRF-KMeans
Input: Set of data points $X = (x_1, \ldots, x_n)$, number of clusters K, set of constraints C, constraint violation costs W, distortion measure D.
Output: Disjoint K-partitioning $(X_1, \ldots, X_K)$ of X such that objective function $\mathcal{J}_{\text{obj}}$ in Eqn. (3.9) is (locally) minimized.
Method:
1. Initialize the K clusters centroids $M^{(0)} = (\mu_1^{(0)}, \ldots, \mu_K^{(0)})$, set t $\leftarrow$ 0
2. Repeat until *convergence*
2a. `E-step`: Given centroids $M^{(t)}$ and distortion parameters $A^{(t)}$, re-assign cluster labels $Y^{(t+1)} = (y_1^{(t+1)}, \ldots, y_n^{(t+1)})$ on X to minimize $\mathcal{J}_{\text{obj}}$.
2b. `M-step(A)`: Given cluster labels $Y^{(t+1)}$ and distortion parameters $A^{(t+1)}$, re-calculate centroids $M^{(t+1)} = (\mu_1^{(t+1)}, \ldots, \mu_K^{(t+1)})$ to minimize $\mathcal{J}_{\text{obj}}$.
2c. `M-step(B)`: Given cluster labels $Y^{(t+1)}$ and centroids $M^{(t+1)}$, re-estimate parameters $A^{(t+1)}$ of the distortion measure to reduce $\mathcal{J}_{\text{obj}}$.
2d. t $\leftarrow$ t+1

Figure 3.3 HMRF-KMeans algorithm

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. If $\lambda > K$, a weighted variant of farthest-first traversal [Hochbaum and 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 and labeled data to obtain cluster representatives that provide a good initial partitioning of the dataset.

3.3.6 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 K-Means 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 NP-hard in any non-trivial HMRF model, similarly to other graphical models such as MRFs and belief networks [Roth, 1996].

greedy ICM assignment

There exist several techniques for computing cluster assignments that approximate the optimal solution in this framework, e.g., iterated conditional modes (ICM) [Besag, 1986, Zhang et al., 2001], belief propagation [Pearl, 1988, Segal et al., 2003], and linear programming relaxation [Kleinberg and Tardos, 1999]. ICM is a greedy strategy that sequentially updates the cluster assignment of each point, keeping the assignments for the other points fixed. In many settings it has comparable performance to more expensive global approximation techniques, but is computationally more efficient; it has been compared with several other approaches by Bilenko and Basu [2004], while in more recent work Lange et al. [2005] have described an alternative efficient method based on the mean-field approximation. 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 $\mathcal{J}_{obj}(x_i, \mu_h)$:

$$\begin{aligned}\mathcal{J}_{\mathrm{obj}}(x_i, \mu_h) = d_A(x_i, \mu_h) &+ \sum_{\substack{(x_i, x_j) \in C_{ML}^i \\ s.t.\ y_i \neq y_j}} w_{ij} \varphi(x_i, x_j) \\ &+ \sum_{\substack{(x_i, x_j) \in C_{CL}^i \\ s.t.\ y_i = y_j}} w_{ij} \big(\varphi^{\max} - \varphi(x_i, x_j)\big) - \log \mathrm{P}(A), \qquad (3.22)\end{aligned}$$

where C_{ML}^i and C_{CL}^i 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 $\mathcal{J}_{\mathrm{obj}}$) along with incurring a minimal penalty for constraint violations caused by this assignment (second and third terms of $\mathcal{J}_{\mathrm{obj}}$). 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 proportionally to their severity, which guides the algorithm towards a desirable partitioning of the data.

3.3.7 M-step

The M-step of the algorithm consists of two parts: centroid re-estimation and distortion measure parameter update.

3.3.7.1 M-Step (A): Centroid Re-estimation

In the first part of the M-step, the cluster centroids M are re-estimated from points currently assigned to them, to decrease the objective function $\mathcal{J}_{\text{obj}}$ in (3.9). 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., 2005a,b]. Additionally, it has been experimentally demonstrated that for clustering with distribution-based measures, e.g., KL divergence, smoothing cluster representatives by a prior using a deterministic annealing schedule leads to considerable improvements [Dhillon and Guan, 2003]. With smoothing controlled by a positive parameter α, each cluster representative μ_h is estimated as follows when d_{I_A} is the distortion measure:

$$\mu_h^{(I_A)} = \frac{1}{1+\alpha}\left(\frac{\sum_{x_i \in X_h} x_i}{|X_h|} + \frac{\alpha}{n}\mathbf{1}\right) \tag{3.23}$$

For directional measures, each cluster representative is the arithmetic mean projected onto unit sphere [Banerjee et al., 2005a]. Taking the distortion parameters into account, centroids are estimated as follows when $d_{\cos_A}$ is the distortion measure:

$$\frac{\mu_h^{(\cos_A)}}{\|\mu_h^{(\cos_A)}\|_A} = \frac{\sum_{x_i \in X_h} x_i}{\|\sum_{x_i \in X_h} x_i\|_A} \tag{3.24}$$

3.3.7.2 M-Step (B): Update of Distortion Parameters

In the second part of the M-step, the parameters of the parameterized distortion measure are updated to decrease the objective function. In general, for parameterized Bregman divergences or directional distances with general parameter priors, it is difficult to attain a closed-form update for the parameters of the distortion measure that can minimize the objective function.[4] Gradient descent provides an alternative avenue for learning the distortion measure parameters.

gradient update for full A

For squared Euclidean distance, a full parameter matrix A is updated during gradient descent using the rule: $A = A + \eta \frac{\partial \mathcal{J}_{euc_A}}{\partial A}$ (where η is the learning rate). Using (3.16), $\frac{\partial \mathcal{J}_{euc_A}}{\partial A}$ can be expressed as:

4. For the specific case of parameterized squared Euclidean distance, a closed-form update of the parameters can be obtained [Bilenko et al., 2004].

$$\frac{\partial \mathcal{J}_{euc_A}}{\partial A} = \sum_{x_i \in X} \frac{\partial d_{euc_A}(x_i, \mu(i))}{\partial A} + \sum_{\substack{(x_i,x_j)\in C_{ML} \\ s.t.\ y_i \neq y_j}} w_{ij} \frac{\partial d_{euc_A}(x_i, x_j)}{\partial A}$$
$$+ \sum_{\substack{(x_i,x_j)\in C_{CL} \\ s.t.\ y_i = y_j}} w_{ij} \left[\frac{\partial \varphi^{\max}_{euc_A}}{\partial A} - \frac{\partial d_{euc_A}(x_i, x_j)}{\partial A} \right] - \frac{\partial \log \mathrm{P}(A)}{\partial A} - n \frac{\partial \log \det(A)}{\partial A}. \tag{3.25}$$

The gradient of the parameterized squared Euclidean distance is given by:

$$\frac{\partial d_{euc_A}(x_i, x_j)}{\partial A} = (x_i - x_j)(x_i - x_j)^T$$

The derivative of the upper bound $\varphi^{\max}_{euc_A}$ is $\frac{\partial \varphi^{\max}_{euc_A}}{\partial A} = \sum_{(x_i,x_j)\in C_{CL}} (x_i - x_j)(x_i - x_j)^T$ if $\varphi^{\max}_{euc_A}$ is computed as described in Section 3.3.3.1.[5]

When Rayleigh priors are used on the set of parameters A, the partial derivative of the log-prior with respect to every individual parameter $a_m \in A$, $\frac{\partial \log \mathrm{P}(A)}{\partial a_m}$, is given by:

$$\frac{\partial \log \mathrm{P}(A)}{\partial a_m} = \frac{1}{a_m} - \frac{a_m}{s^2} \tag{3.26}$$

The gradient of the distortion normalizer $\log \det(A)$ term is as follows:

$$\frac{\partial \log \det(A)}{\partial A} = 2A^{-1} - diag(A^{-1}). \tag{3.27}$$

gradient update for diagonal A

For parameterized cosine distance and KL divergence, a diagonal parameter matrix A is considered, where $a = diag(A)$ is a vector of positive weights. During gradient descent, each weight a_m is individually updated as: $a_m = a_m + \eta \frac{\partial \mathcal{J}_{\mathrm{obj}}}{\partial a_m}$ (η is the learning rate). Using (3.14), $\frac{\partial \mathcal{J}_{\mathrm{obj}}}{\partial a_m}$ can be expressed as:

$$\frac{\partial \mathcal{J}_{\mathrm{obj}}}{\partial a_m} = \sum_{x_i \in X} \frac{\partial d_A(x_i, \mu(i))}{\partial a_m} + \sum_{\substack{(x_i,x_j)\in C_{ML} \\ s.t.\ y_i \neq y_j}} w_{ij} \frac{\partial \varphi(x_i, x_j)}{\partial a_m}$$
$$+ \sum_{\substack{(x_i,x_j)\in C_{CL} \\ s.t.\ y_i = y_j}} w_{ij} \left[\frac{\partial \varphi^{\max}}{\partial a_m} - \frac{\partial \varphi(x_i, x_j)}{\partial a_m} \right] - \frac{\partial \log \mathrm{P}(A)}{\partial a_m} \tag{3.28}$$

5. In practice, one can initialize $\varphi^{\max}_{euc_A}$ with a sufficiently large constant, which would make its derivative zero. Accordingly, an extra condition must be then inserted into the algorithm to guarantee that penalties for violated cannot-link constraints are never negative, in which case the constant must be increased.

Calculation of the gradient $\frac{\partial \mathcal{J}_{\text{obj}}}{\partial a_m}$ for cosine distance and KL divergence, which are parameterized by a diagonal matrix A, needs the gradients of the corresponding distortion measures and constraint potential functions, which are:

$$\begin{aligned}
\frac{\partial d_{\cos_A}(x_i,x_j)}{\partial a_m} &= \frac{x_{im}x_{jm}\|x_i\|_A\|x_j\|_A - x_i^T A x_j \frac{x_{im}^2\|x_j\|_A^2 + x_{jm}^2\|x_i\|_A^2}{2\|x_i\|_A\|x_j\|_A}}{\|x_i\|_A^2\|x_j\|_A^2}, \\
\frac{\partial d_{I_A}(x_i,x_j)}{\partial a_m} &= x_{im}\log\frac{x_{im}}{x_{jm}} - (x_{im}-x_{jm}), \\
\frac{\partial d_{IM_A}(x_i,x_j)}{\partial a_m} &= x_{im}\log\frac{2x_{im}}{x_{im}+x_{jm}} + x_{jm}\log\frac{2x_{jm}}{x_{im}+x_{jm}},
\end{aligned} \tag{3.29}$$

while the gradient of the upper bound $\frac{\partial \varphi^{\max}}{\partial a_m}$ is 0 for parameterized cosine and 1 for parameterized KL divergence, as follows from the expressions for these constants in Sections 3.3.3.2 and 3.3.3.3.

Overall, the distance learning step results in modifying the distortion measure so that data points in violated must-link constraints are brought closer together, while points in violated cannot-link constraints are pulled apart. This process leads to a transformed data space that facilitates partitioning of the unlabeled data, by attempting to mend the constraint violations as well as reflecting the natural variance in the data. See chapters 14-16 for several alternative techniques that change the data representation leading to better estimates of similarity between data points.

3.3.8 Convergence of HMRF-KMeans

The HMRF-KMeans algorithm alternates between updating the assignment of points to clusters, and updating the parameters. Since all updates ensure a decrease in the objective function, each iteration of HRMF-KMeans monotonically decreases the objective function. 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 (3.14). 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:

$$d_A(x_i,\mu(i)) + \sum_{\substack{(x_i,x_j)\in C_{ML}^i \\ s.t.\ y_i\neq y_j}} w_{ij}\varphi(x_i,x_j) + \sum_{\substack{(x_i,x_j)\in C_{CL}^i \\ s.t.\ y_i=y_j}} w_{ij}\big(\varphi^{\max} - \varphi(x_i,x_j)\big) - \log \mathrm{P}(A).$$

Given a set of centroids and distortion parameters, the new cluster assignment of points will decrease $\mathcal{J}_{\text{obj}}$ or keep it unchanged.

For analyzing the centroid re-estimation step, let us consider an equivalent form of (3.14):

$$\mathcal{J}_{\text{obj}} = \sum_{h=1}^{K} \sum_{x_i \in X_h} d_A(x_i, \mu_h) + \sum_{\substack{(x_i,x_j) \in C^i_{ML} \\ s.t.\ y_i \neq y_j}} w_{ij} \varphi(x_i, x_j)$$
$$+ \sum_{\substack{(x_i,x_j) \in C^i_{CL} \\ s.t.\ y_i = y_j}} w_{ij} \big(\varphi^{\max} - \varphi(x_i, x_j)\big) - \log \mathrm{P}(A), \tag{3.30}$$

Each cluster centroid μ_h is re-estimated by taking the mean of the points in the partition X_h, which minimizes the component $\sum_{x_i \in X_h} d_A(x_i, \mu_h)$ of $\mathcal{J}_{\text{obj}}$ in (3.30) contributed by the partition X_h. The constraint potential and the prior term in the objective function do not take a part in centroid re-estimation, because they are not explicit functions of the centroid. So, given the cluster assignments and the distortion parameters, $\mathcal{J}_{\text{obj}}$ will decrease or remain the same in this step.

For the parameter estimation step, the gradient-descent update of the parameters in M-step (B) decreases $\mathcal{J}_{\text{obj}}$ or keeps it unchanged. Hence the objective function decreases after every cluster assignment, centroid re-estimation and parameter 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, $\mathcal{J}_{\text{obj}}$ 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 due to the fact that A is positive definite. Since $\mathcal{J}_{\text{obj}}$ is bounded below, and HMRF-KMeans results in a decreasing sequence of objective function values, the value sequence must have a limit. The limit in this case will be a fixed point of $\mathcal{J}_{\text{obj}}$ since neither updating the assignments or the parameters can further decrease the value of the objective function. As a result, the HMRF-KMeans algorithm will converge to a fixed point of the objective. In practice, convergence can be determined if subsequent iterations of HMRF-KMeans result in insignificant changes in $\mathcal{J}_{\text{obj}}$.

3.4 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 section an active learning scheme for the HMRF model is presented, which can improve clustering performance with as few queries as possible. Formally, the scheme has access to a (noiseless) oracle that can assign a must-link or cannot-link label on a given pair (x_i, x_j), and it can pose a constant number of queries to the oracle.[6]

In order to get pairwise constraints that are more informative than random in

6. The oracle can also give a *don't-know* response to a query, in which case that response is ignored (pair not considered as a constraint) and that query is not posed again later.

farthest first traversal

the HMRF model, an active learning scheme for selecting pairwise constraints using the *farthest-first* traversal scheme is developed. 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 and Shmoys, 1985], and has also been used to construct hierarchical clusterings with performance guarantees at each level of the hierarchy [Dasgupta, 2002].

good initialization for K-Means

Basu et al. [2002] observed that initializing K-Means with centroids estimated from a set of labeled examples for each cluster gives significant performance improvements. Under certain generative model-based assumptions, one can connect the mixture of Gaussians model to K-Means with squared Euclidean distance [Kearns et al., 1997]. A direct calculation using Chernoff bounds shows that if a particular cluster with an underlying Gaussian model is seeded with points drawn independently at random from the corresponding Gaussian distribution, the deviation of the centroid estimates falls exponentially with the number of seeds; hence seeding results in good initial centroids. Since good initial centroids are very critical for the success of greedy algorithms such as K-Means, the same principle is followed for the pairwise case: the goal 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. The proposed active learning scheme has two phases: Explore and Consolidate, which are discussed next.

Algorithm: Explore
Input: Set of data points $X = (x_1, \ldots, x_n)$, access to an oracle that answers pairwise queries, number of clusters K, total number of queries Q.
Output: $\lambda \leq K$ disjoint neighborhoods $N = (N_1, \ldots, N_\lambda)$ 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 existing neighborhoods N
    if, by querying, it is found that x is cannot-linked to all existing neighborhoods
        $\lambda \leftarrow \lambda + 1$, start a new neighborhood N_λ with x
    else
        add x to the neighborhood with which it is must-linked

Figure 3.4 Algorithm Explore

3.4.1 Exploration

The EXPLORE 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. Our algorithm EXPLORE (Figure 3.4) 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. In the latter case, active learning enters the consolidation phase.

form skeleton of neighborhoods

Algorithm: CONSOLIDATE
Input: Set of data points $X = (x_1, \ldots, x_n)$, 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 $(\mu_1, \ldots, \mu_K)$ 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 3.5 Algorithm CONSOLIDATE

3.4.2 Consolidation

The basic idea in CONSOLIDATE (Figure 3.5) is as follows: since there is at least one labeled point from all 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 it can be inferred 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.

consolidate neighborhoods

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 dataset, 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), while samples obtained using farthest-first traversal may not have such properties.

3.5 Experimental Results

This section describes the experiments that were performed to demonstrate the effectiveness of various aspects of HMRF-KMeans.

3.5.1 Datasets

Experiments were run on both low-dimensional and high-dimensional datasets to evaluate the HMRF-KMeans framework with different distortion measures. For the low-dimensional datasets, on which squared Euclidean distance was used as the distortion measure, the following datasets were considered:

low-dimensional datasets

- Three datasets from the UCI repository: *Iris*, *Wine*, and *Ionosphere* [Blake and Merz, 1998];
- The *Protein* dataset used by Xing et al. [2003] and Bar-Hillel et al. [2003];
- Randomly sampled subsets from the *Digits* and *Letters* handwritten character recognition datasets, also from the UCI repository. For *Digits* and *Letters*, two sets of three classes were chosen: {**I, J, L**} from *Letters* and {**3, 8, 9**} from *Digits*, sampling 10% of the data points from the original datasets randomly. These classes were chosen since they represent difficult visual discrimination problems.

Table 3.1 summarizes the properties of the low-dimensional datasets: the number of instances, the number of dimensions, and the number of classes.

Table 3.1 Low-dimensional datasets used in experimental evaluation

	Iris	*Wine*	*Ionosphere*	*Protein*	*Letters*	*Digits*
Instances	150	178	351	116	227	317
Dimensions	4	13	34	20	16	16
Classes	3	3	2	6	3	3

For the high-dimensional text data, 3 datasets that have the characteristics of being sparse, high-dimensional, and having a small number of points compared to the dimensionality of the space were considered. This is done for two reasons:

- When clustering sparse high-dimensional data, e.g., text documents represented using the vector space model, it is particularly difficult to cluster small datasets, as observed by Dhillon and Guan [2003]. The purpose of performing experiments on these subsets is to scale down the sizes of the datasets for computational reasons but at the same time not scale down the difficulty of the tasks.
- Clustering small number of sparse high-dimensional data points is a likely scenario in realistic applications. For example, when clustering the search results in a web-search engine like Vivísimo[7], typically the number of webpages that are being clustered is in the order of hundreds. However the dimensionality of the 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. On such datasets, clustering algorithms can easily get stuck in local optima: in such cases 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 and Guan, 2003]. Supervision in the form of pairwise constraints is most beneficial in such cases and may significantly improve clustering quality.

high-dimensional datasets

Three datasets were derived from the *20-Newsgroups* collection.[8] This collection has messages harvested from 20 different Usenet newsgroups, 1000 messages from each newsgroup. From the original dataset, a reduced dataset was created by taking a random subsample of 100 documents from each of the 20 newsgroups. Three datasets were created by selecting 3 categories from the reduced collection. *News-Similar-3* consists of 3 newsgroups on similar topics (`comp.graphics`, `comp.os.ms-windows`, `comp.windows.x`) with significant overlap between clusters due to cross-posting. *News-Related-3* consists of 3 newsgroups on related topics (`talk.politics.misc`, `talk.politics.guns`, and `talk.politics.mideast`). *News-Different-3* consists of articles posted in 3 newsgroups that cover different topics (`alt.atheism`, `rec.sport.baseball`, `sci.space`) with well-separated clusters. All the text datasets were converted to the vector-space model by tokenization, stop-word removal, TF-IDF weighting, and removal of very high-frequency and low-frequency words, following the methodology of Dhillon and Modha [2001].

Table 3.2 summarizes the properties of the high-dimensional datasets.

7. http://www.vivisimo.com
8. http://www.ai.mit.edu/people/jrennie/20Newsgroups

Table 3.2 High-dimensional datasets used in experimental evaluation

	News-Different-3	*News-Related-3*	*News-Similar-3*
Instances	300	300	300
Dimensions	3251	3225	1864
Classes	3	3	3

3.5.2 Clustering Evaluation

Normalized mutual information (NMI) was used as the clustering evaluation measure. NMI is an external clustering validation metric that estimates the quality of the clustering with respect to a given underlying class labeling of the data: it measures how closely the clustering algorithm could reconstruct the underlying label distribution in the data [Strehl et al., 2000]. If $\hat{Y}$ is the random variable denoting the cluster assignments of the points and Y is the random variable denoting the underlying class labels on the points, then the NMI measure is defined as:

$$NMI = \frac{I(Y;\hat{Y})}{(H(Y) + H(\hat{Y}))/2} \tag{3.31}$$

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 and Thomas, 1991]. NMI effectively measures the amount of statistical information shared by the random variables representing the cluster assignments and the user-labeled class assignments of the data points. Though various clustering evaluation measures have been used in the literature, NMI and it's variants have become popular lately among clustering practitioners [Dom, 2001, Fern and Brodley, 2003, Meila, 2003].

3.5.3 Methodology

Learning curves were generated using two-fold cross-validation performed over 20 runs on each dataset. In every trial, 50% of the dataset was set aside as the training fold. Every point on the learning curve corresponds to the number of constraints on pairs of data points from the training fold. These constraints are obtained by randomly selecting pairs of points from the training fold 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 datasets did not provide individual weights for the constraints. The gradient step size η for learning the distortion measure parameters and the Rayleigh prior width parameter s were set based on pilot studies. The gradient step size was set to $\eta = 100.0$ for clustering with weighted cosine distance $d_{\cos_A}$ and $\eta = 0.08$ for weighted I divergence d_{I_A}. The Rayleigh prior width parameter was set to $s = 1$. In a real-life setting, the free parameters of the algorithm could be tuned using cross-validation with a hold-out set. The clustering algorithm was run on the whole dataset, but NMI was calculated using points in the test fold.

Sensitivity experiments were performed with HMRF-KMeans to study the effectiveness of each component of the algorithm. The proposed HMRF-KMeans algorithm was compared with three ablations, as well as with unsupervised K-Means clustering. The following variants were compared for distortion measures $d_{\cos_A}$, d_{I_A} and d_{euc_A}:

sensitivity experiments

- KMeans-C-D-R is the complete HMRF-KMeans algorithm that incorporates constraints in cluster assignments (C) as described in Section 3.3.6, includes distance learning (D) as described in Section 3.3.7, and also performs regularization (R) using a Rayleigh prior as described in Section 3.3.2;
- KMeans-C-D is the first ablation of HMRF-KMeans that includes all components except for regularization of distortion measure parameters;
- KMeans-C is an ablation of HMRF-KMeans that uses pairwise supervision for initialization and cluster assignments, but does not perform distortion measure learning;
- KMeans is the unsupervised K-Means algorithm.

The goal of these experiments was to evaluate the utility of each component of the HMRF framework and identify settings in which particular components are beneficial.

3.5.4 Results and Discussion

Low-dimensional datasets: Figures 3.6-3.11 show learning curves for the ablation experiments on the six low-dimensional datasets. Across all datasets, the overall HMRF-KMeans approach without regularization (KMeans-C-D) outperforms the constraints-only ablation and unsupervised KMeans. Since the performance of KMeans-C-D-R is not substantially different from KMeans-C-D, it can be concluded that regularization does not lead to performance improvements on low-dimensional datasets. This can be explained by the fact that the number of distortion measure parameters is small for low-dimensional domains while estimates obtained from data do not have high variance, and therefore incorporating a prior in the probabilistic model is not necessary.

For the *Wine*, *Protein*, and *Digits-389* datasets, the difference between ablations that utilize metric learning (KMeans-C-D-R and KMeans-C-D) and those that do not (KMeans-C and KMeans) at the beginning of the learning curve indicates that even in the absence of constraints, weighting features by their variance (essentially using unsupervised Mahalanobis distance) improves clustering accuracy. For the *Wine* dataset, additional constraints provide an insubstantial improvement in cluster quality on this dataset, which shows that meaningful feature weights are obtained from scaling by variance using just the unlabeled data.

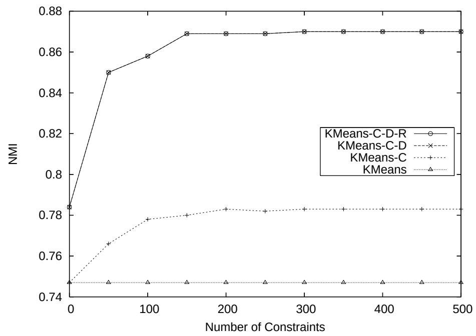

Figure 3.6 Results for d_{euc} on *Iris*

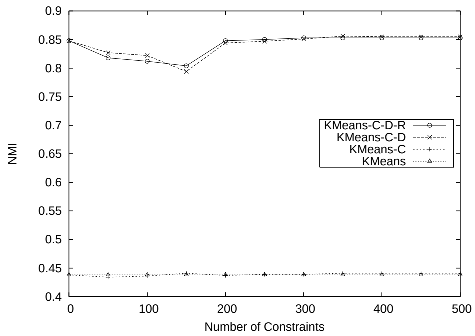

Figure 3.7 Results for d_{euc} on *Wine*

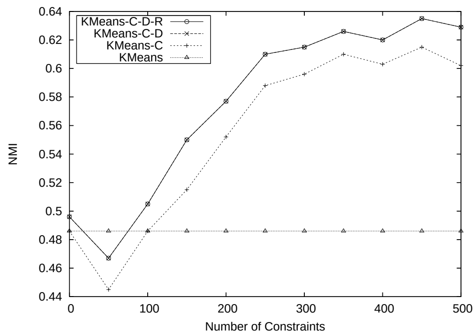

Figure 3.8 Results for d_{euc} on *Protein*

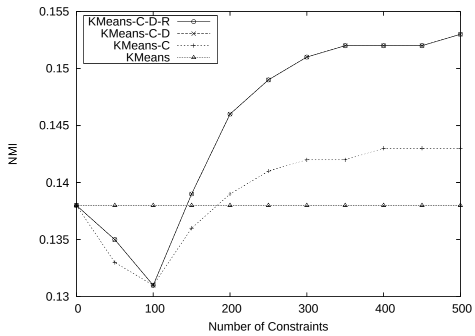

Figure 3.9 Results for d_{euc} on *Ionosphere*

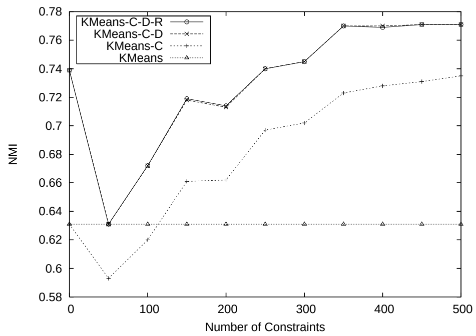

Figure 3.10 Results for d_{euc} on *Digits-389*

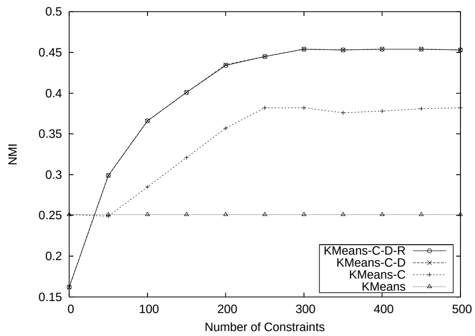

Figure 3.11 Results for d_{euc} on *Letters-IJL*

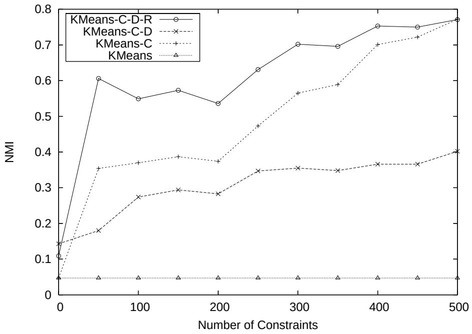

Figure 3.12 Results for $d_{\cos_A}$ on *News-Different-3*

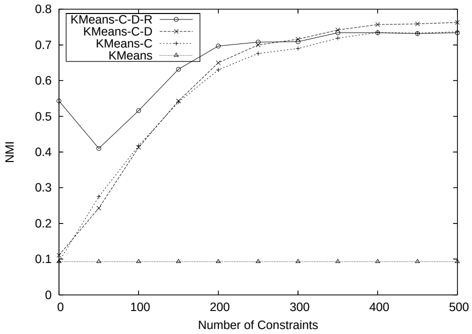

Figure 3.13 Results for d_{I_A} on *News-Different-3*

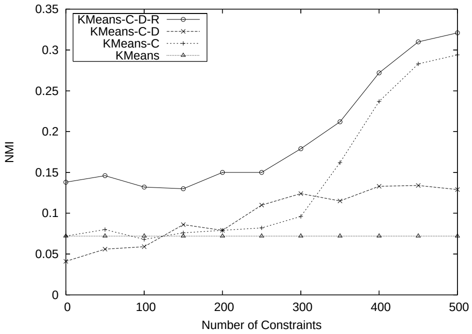

Figure 3.14 Results for $d_{\cos_A}$ on *News-Related-3*

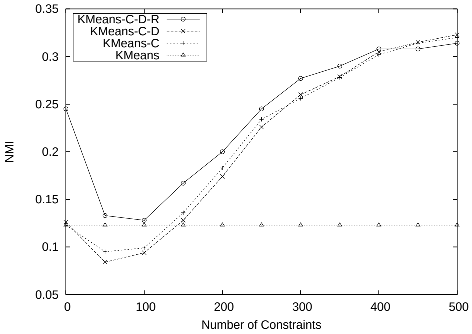

Figure 3.15 Results for d_{I_A} on *News-Related-3*

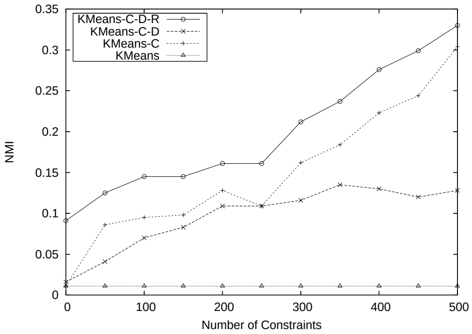

Figure 3.16 Results for $d_{\cos_A}$ on *News-Similar-3*

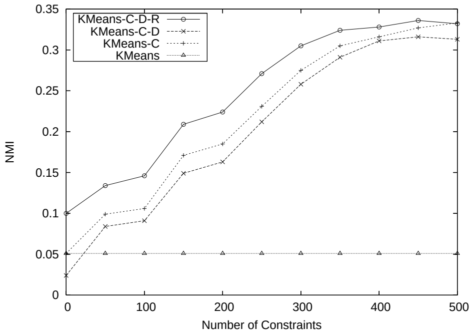

Figure 3.17 Results for d_{I_A} on *News-Similar-3*

Some of the metric learning curves display a characteristic "dip", where clustering accuracy decreases as a few initial constraints are provided, but after a certain point starts to increase and eventually rises above the initial point on the learning curve. One possible explanation of this phenomenon is that metric parameters learned using too few constraints are unreliable, and a significant number of constraints is required by the metric learning mechanism to estimate parameters accurately. Overall, when both constraints and distortion measure learning are utilized, the unified approach benefits from the individual strengths of the two methods, as can be seen from the KMeans-C-D results.

High-dimensional datasets: Figures 3.12, 3.14 and 3.16 present the results for the ablation experiments where weighted cosine similarity $d_{\cos_A}$ was used as the distortion measure, while Figures 3.13, 3.15 and 3.17 summarize experiments where weighted I divergence d_{I_A} was used.

As the results demonstrate, the full HMRF-KMeans algorithm with regularization (KMeans-C-D-R) outperforms the unsupervised K-Means baseline as well as the ablated versions of the algorithm for both distortion measures $d_{\cos_A}$ and d_{I_A}. As can be seen from results for zero pairwise constraints in Figs. 3.12-3.17, distortion measure learning is beneficial even in the absence of any pairwise constraints, since it allows capturing the relative importance of the different attributes in the unsupervised data. In the absence of supervised data or when no constraints are violated, distance learning attempts to minimize the objective function by adjusting the weights given the distortion between the unsupervised datapoints and their corresponding cluster representatives.

For high-dimensional datasets, regularization is clearly beneficial to performance, as can be seen from the improved performance of KMeans-C-D-R over KMeans-C-D on all datasets. This can be explained by the fact that the number of distortion measure parameters is large for high-dimensional datasets, and therefore algorithm-based estimates of parameters tend to be unreliable unless they incorporate a prior.

Overall, these results show that the HMRF-KMeans algorithm effectively incorporates labeled and unlabeled data in all its stages, each of which improves the clustering quality.

3.6 Related Work

The problem of integrating limited supervision in clustering algorithms has been studied by a number of authors in recent work. Early approaches to semi-supervised clustering relied on incorporating penalties for violating constraints into the objective function, leading to algorithms that avoid clusterings in which constraints are not satisfied. COP-KMeans is one such method where constraint violations are explicitly avoided in the assignment step of the K-Means algorithm [Wagstaff et al., 2001, Wagstaff, 2002]. Another method proposed by Demiriz et al. [1999] utilizes genetic algorithms to optimize an objective function that combines cluster compactness and cluster purity and that decreases with constraint violations.

In subsequent work, several approaches have been proposed that consider semi-supervised clustering within a probabilistic framework. Segal et al. [2003] describe a model for semi-supervised clustering with constraints that combines a binary Markov network derived from pairwise protein interaction data and a Naive Bayes Markov network modeling gene expression data. Another probabilistic approach described by Shental et al. [2004] incorporates must-link constraints via modeling them as *chunklets*, sets of points known to belong to the same class, while cannot-link constraints are utilized via potentials in a binary Markov network. HMRFs have previously been used for image segmentation by Zhang et al. [2001], who have also described an EM-based clustering algorithm. More recently, Lange et al. [2005] proposed an approach that incorporates labeled and unlabeled data within an HMRF-like model, while a mean field approximation method for posterior inference is used in the E-step of the algorithm. The HMRF framework described in this chapter differs from these approaches in that it explicitly incorporates learning of the distortion measure parameters within the clustering algorithm and facilitates the use of diverse distance measures; however, a number of the proposed methods could be integrated within the HMRF framework.

Spectral clustering methods—algorithms that perform clustering by decomposing the pairwise affinity matrix derived from data—have been increasingly popular recently [Weiss, 1999, Ng et al., 2002], and several semi-supervised approaches have been developed within the spectral clustering framework. Kamvar et al. [2003] have proposed directly injecting the constraints into the affinity matrix before subsequent clustering, while De Bie et al. [2004] reformulated the optimization problem corresponding to spectral clustering by incorporating a separate label constraint matrix. Additionally, spectral clustering methods can be viewed as variants of the graph-cut approaches to clustering [Shi and Malik, 2000], a connection that motivated the *correlation clustering* method proposed by [Bansal et al., 2004], where the constraints correspond to edge labels between vertices representing datapoints.

Another family of semi-supervised clustering methods has focused on modifying the distance function employed by the clustering algorithm. In early work, Cohn et al. [2003] proposed using a weighted variant of Jensen-Shannon divergence within the EM clustering algorithm, with the weights learned using gradient descent based on constraint violations. Within the family of hierarchical agglomerative clustering algorithms, Klein et al. [2002] proposed modifying the squared Euclidean distance using the shortest-path algorithm. Several researchers have proposed methods for learning the parameters of the weighted Mahalanobis distance, a generalization of Euclidean distance, within the context of semi-supervised clustering. Xing et al. [2003] utilized convex optimization and iterative projections to learn the weight matrix of Mahalanobis distance within K-Means clustering. Another approach focused on parameterized Mahalanobis distance is the Relevant Component Analysis (RCA) algorithm proposed by Bar-Hillel et al. [2003], where convex optimization is also used to learn the weight matrix.

Learning distance metrics within semi-supervised clustering relates to a large set of approaches for transforming the data representation to make it more suitable to a particular learning task. Within this volume, chapters 14-16 describe several advanced techniques for changing the geometry of the data space to obtain better estimates of similarity between data points; integrating these methods with clustering algorithms provides a number of promising avenues for future work.

3.7 Conclusions

In this chapter, a generative probabilistic framework for semi-supervised clustering has been introduced. It relies on Hidden Random Markov Fields (HMRFs) to utilize both unlabeled data and supervision in the form of pairwise constraints during the clustering process. The framework can be used with a number of distortion (distance) measures, including Bregman divergences and directional measures, and it facilitates training the distance parameters to adapt to specific datasets.

An algorithm HMRF-KMeans for performing clustering in this framework has been presented that incorporates pairwise supervision in different stages of the clustering: initialization, cluster assignment, and parameter estimation. Three particular instantiations of the algorithm, based on different distortion measures, have been discussed: squared Euclidean distance, which is common for clustering low-dimensional data, and KL divergence and cosine distance, which are popular for clustering high-dimensional directional data. Finally, a new method has been presented for acquiring supervision from a user in the form of effective pairwise constraints for semi-supervised clustering – such an active learning algorithm would be useful in an interactive query-driven clustering framework.

The HMRF model can be viewed as a unification of constrained-based and distance-based semi-supervised clustering approaches. It can be expanded to a more general setting where every cluster has a corresponding distinct distortion measure [Bilenko et al., 2004], leading to a clustering algorithm that can identify clusters of different shapes. Empirical evaluation of the framework described in this chapter can be found in several previous publications: active learning experiments are discussed in [Basu et al., 2004a], while [Bilenko et al., 2004] and [Basu et al., 2004b] contain results for low-dimensional and high-dimensional datasets respectively, and [Bilenko and Basu, 2004] compares several approximate inference methods for E-Step discussed in Section 3.3.6.

An important practical issue in using generative models for SSL is model selection. For semi-supervised clustering with constraints, the key model selection issue is one of choosing the right number of clusters. One can consider using a traditional model selection criterion suitable for the supervised setting, or perform model selection by cross-validation. An alternative is to perform model-selection using bounds on the test-set error-rate such that valuable supervised data is saved for learning. The PAC-MDL bounds [Blum and Langford, 2003] provide such a tool that has been successfully applied to model selection for clustering [Banerjee et al., 2005a], and can be readily extended to the semi-supervised clustering setting. In fact, the semi-supervised clustering setting is more natural since PAC-MDL bounds are applicable for transductive learning. Alternative methods of model selection are a good topic for future research.

References

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

A. Banerjee, I. Dhillon, J. Ghosh, and S. Sra. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research*, 6:1345–1382, 2005a.

A. Banerjee, S. Merugu, I. Dhilon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005b.

N. Bansal, A. L. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1–3):89–113, 2004.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *Proceedings of ICML*, pages 11–18, Washington, DC, 2003.

S. Basu, A. Banerjee, and R. J. Mooney. Semi-supervised clustering by seeding. In *Proceedings of ICML*, pages 19–26, 2002.

S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of SIAM SDM*, 2004a.

S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of ACM SIGKDD*, pages 59–68, Seattle, WA, 2004b.

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

M. Bilenko and S. Basu. A comparison of inference techniques for semi-supervised clustering with hidden markov random fields. In *Proceedings of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields (SRL-2004)*, Banff, Canada, 2004.

M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of ICML*, pages 81–88, Banff, Canada, 2004.

M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of ACM SIGKDD*, pages 39–48, Washington, DC, 2003.

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

A. L. Blum and J. C. Langford. PAC-MDL bounds. In *Proceedings of COLT*, 2003.

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

D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback. Technical Report TR2003-1892, Cornell University, 2003.

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

S. Dasgupta. Performance guarantees for hierarchical clustering. In *Proceedings of COLT*, pages 351–363, 2002.

T. De Bie, J. A. K. Suykens, and B. De Moor. Learning from general label constraints. In *Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition*, pages 671–679. Lisbon, Portugal, 2004.

A. Demiriz, K. P. Bennett, and M. J. Embrechts. Semi-supervised clustering using genetic algorithms. In *Proceedings of ANNIE*, pages 809–814, 1999.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *JRSSB*, 39:1–38, 1977.

I. S. Dhillon and Y. Guan. Information theoretic clustering of sparse co-occurrence data. In *Proceedings of ICDM*, pages 517–521, 2003.

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

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

X. Fern and C. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of ICML*, 2003.

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

J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. Unpublished manuscript, 1971.

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

S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *Proceedings of IJCAI*, pages 561–566, Acapulco, Mexico, 2003.

M. Kearns, Y. Mansour, and A. Y. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Proceedings of UAI*, pages 282–293, 1997.

D. Klein, S. D. Kamvar, and C. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of ICML*, pages 307–314, Sydney, Australia, 2002.

J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In *Proceedings of FOCS*, pages 14–23, 1999.

T. Lange, M. H. C. Law, A. K. Jain, and J. M. Buhmann. Learning with constrained and unlabeled data. In *CVPR*, pages 731–738. San Diego, CA, 2005.

J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

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

K. V. Mardia and P. E. Jupp. *Directional Statistics*. John Wiley and Sons, 2nd edition, 2000.

A. McCallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *Proceedings of ICML*, Madison, WI, 1998.

M. Meila. Comparing clusterings by the variation of information. In *Proceedings of COLT*, pages 173–187, 2003.

R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. MIT Press, 1998.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS 14*. 2002.

A. Papoulis and S. U. Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Inc., New York, 4th edition, 2001.

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

F. C. N. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. In *ACL*, pages 183–190, Columbus, Ohio, 1993.

D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996.

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

N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In *NIPS 16*. 2004.

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

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

K. Wagstaff. *Intelligent Clustering with Instance-Level Constraints*. PhD thesis, Cornell University, 2002.

K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K-Means clustering with background knowledge. In *Proceedings of ICML*, pages 577–584, 2001.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. Technical Report 649, Department of Statistics, University of California, Berkeley, 2003.

Y. Weiss. Segmentation using eigenvectors: a unifying view. In *ICCV*, pages 975–982. Kerkyra, Greece, 1999.

E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *NIPS 15*, pages 505–512, 2003.

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