Contents

0-8493-0052-5/00/$0.00+$.50

List of Tables

0-8493-0052-5/00/$0.00+$.50

List of Figures

0-8493-0052-5/00/$0.00+$.50

Notation and Symbols

Sets of Numbers	
$\mathbb{N}$	the set of natural numbers, $\mathbb{N} = \{1, 2, \dots\}$
$\mathbb{R}$	the set of reals
$[n]$	compact notation for $\{1, \dots, n\}$
$x \in [a, b]$	interval $a \leq x \leq b$
$x \in (a, b]$	interval $a < x \leq b$
$x \in (a, b)$	interval $a < x < b$
$\|C\|$	cardinality of a set C (for finite sets, the number of elements)
Data	
$\mathcal{X}$	the input domain
d	(used if $\mathcal{X}$ is a vector space) dimension of $\mathcal{X}$
m	number of underlying classes in the labeled data
k	number of clusters (can be different from m)
l, u	number of labeled, unlabeled training examples
n	total number of examples, $n = l + u$.
i, j	indices, often running over $[n]$ or $[k]$
x_i	input data point $x_i \in \mathcal{X}$
y_j	output cluster label $y_j \in [K]$
X	a sample of input data points, $X = (x_1, \dots, x_n)$ and $X = \{X_l \cup X_u\}$
Y	output cluster labels, $Y = (y_1, \dots, y_n)$ and $Y = \{Y_l \cup Y_u\}$
Π_X	k block clustering (set partition) on X: $\{\pi_1, \pi_2 \dots \pi_k\}$
$D(x, y)$	distance between points x and y
X_l	labeled part of X, $X_l = (x_1, \dots, x_l)$
Y_l	part of Y where labels are specified, $Y_l = (y_1, \dots, y_l)$
X_u	unlabeled part of X, $X_u = (x_{l+1}, \dots, x_{l+u})$
Y_u	part of Y where labels are not specified, $Y_u = (y_{l+1}, \dots, y_{l+u})$
C	set of constraints
W	weights on constraints
$C_=$	conjunction of must-link constraints
$C_\neq$	conjunction of cannot-link constraints
$c_=(i, j)$	must-link constraint between x_i and x_j
$c_\neq(i, j)$	cannot-link constraint between x_i and x_j
$w_=(i, j)$	weight on must-link constraint $c_=(i, j)$
$w_\neq(i, j)$	weight on cannot-link constraint $c_\neq(i, j)$

Kernels

$\mathcal{H}$	feature space induced by a kernel
Φ	feature map, $\Phi : \mathcal{X} \to \mathcal{H}$
K	kernel matrix or Gram matrix, $K_{ij} = k(x_i, x_j)$

Vectors, Matrices and Norms

$\mathbf{1}$	vector with all entries equal to one
$\mathbf{I}$	identity matrix
$A^\top$	transposed matrix (or vector)
A^{-1}	inverse matrix (in some cases, pseudo-inverse)
$\mathbf{tr}\,(A)$	trace of a matrix
$\mathbf{det}\,(A)$	determinant of a matrix
$\langle \mathbf{x}, \mathbf{x}' \rangle$	dot product between $\mathbf{x}$ and $\mathbf{x}'$
$\|\cdot\|$	2-norm, $\|\mathbf{x}\| := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$
$\|\cdot\|_p$	p-norm , $\|\mathbf{x}\|_p := \left(\sum_{i=1}^N |x_i|^p \right)^{1/p}$, $N \in \mathbb{N} \cup \{\infty\}$
$\|\cdot\|_\infty$	∞-norm , $\|\mathbf{x}\|_\infty := \sup_{i=1}^N |x_i|$, $N \in \mathbb{N} \cup \{\infty\}$

Functions

ln	logarithm to base e
$\log_2$	logarithm to base 2
f	a function, often from $\mathcal{X}$ or $[n]$ to $\mathbb{R}$, $\mathbb{R}^M$ or $[M]$
$\mathcal{F}$	a family of functions
$L_p(\mathcal{X})$	function spaces, $1 \leq p \leq \infty$

Probability

$\mathrm{P}\{\cdot\}$	probability of a logical formula
$\mathrm{P}(C)$	probability of a set (event) C
$p(x)$	density evaluated at $x \in \mathcal{X}$
$\mathbf{E}\,[\cdot]$	expectation of a random variable
$\mathbf{Var}\,[\cdot]$	variance of a random variable
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with mean μ and variance σ^2

Graphs	
g	graph $\mathsf{g} = (V, E)$ with nodes V and edges E
$\mathcal{G}$	set of graphs
$\mathbf{W}$	weighted adjacency matrix of a graph ($\mathbf{W}_{ij} \neq 0 \Leftrightarrow (i,j) \in E$)
$\mathbf{D}$	(diagonal) degree matrix of a graph, $\mathbf{D}_{ii} = \sum_j W_{ij}$
$\mathcal{L}$	normalized graph Laplacian, $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$
L	un-normalized graph Laplacian, $L = \mathbf{D} - \mathbf{W}$
Miscellaneous	
I_A	characteristic (or indicator) function on a set A i.e., $I_A(x) = 1$ if $x \in A$ and 0 otherwise
δ_{ij}	Kronecker δ ($\delta_{ij} = 1$ if $i = j, 0$ otherwise)
δ_x	Dirac δ, satisfying $\int \delta_x(y) f(y) dy = f(x)$
$O(g(n))$	a function $f(n)$ is said to be $O(g(n))$ if there exist constants $C > 0$ and $n_0 \in \mathbb{N}$ such that $\lvert f(n)\rvert \leq Cg(n)$ for all $n \geq n_0$
$o(g(n))$	a function $f(n)$ is said to be $o(g(n))$ if there exist constants $c > 0$ and $n_0 \in \mathbb{N}$ such that $\lvert f(n)\rvert \geq cg(n)$ for all $n \geq n_0$
rhs/lhs	shorthand for "right/left hand side"
■	the end of a proof

Part I

Part A

Chapter 1

Pairwise Constraints as Priors in Probabilistic Clustering

Zhengdong Lu
Department of Computer Science and Engineering
OGI School of Science and Engineering , OHSU
Beaverton, OR 97006

Todd K. Leen
Department of Computer Science and Engineering
OGI School of Science and Engineering , OHSU
Beaverton, OR 97006

1.1 Introduction

While clustering is usually executed completely unsupervised, there are circumstances in which we have prior belief (with varying degrees of certainty) that pairs of samples should (or should not) be assigned to the same cluster. More specifically, we specify two types of pairwise relations:

- **must-link**: two samples should be assigned to the same cluster
- **cannot-link**: two samples should be assigned to different clusters.

We use $C_=$ and $C_{\neq}$ to denote the set of must-links and cannot-links.

Our interest in such problems was kindled when we tried to manually segment a satellite image by grouping small image clips from the image. One finds that it is often hard to assign the image clips to different "groups" since we do not know clearly the characteristic of each group, or even how many classes we should have. In contrast, it is much easier to compare two image clips and to decide how much they look alike and thus how likely they should be in one cluster. Another example is in information retrieval. Cohn et. al. [6] suggested that in creating a document taxonomy, the expert critique is often in the form "these two documents shouldn't be in the same cluster". The last example is continuity, which suggests that neighboring pairs of samples in a time series or in an image are likely to belong to the same class of object,

0-8493-0052-5/00/$0.00+$.50

is also a source of clustering preferences [19, 1]. We would like these preferences to be incorporated into the cluster structure so that the assignment of out-of-sample data to clusters captures the concept(s) that give rise to the preferences expressed in the training data.

Some work has been done on adopting traditional clustering methods, such as K-means, to incorporate pairwise relations [21, 2, 10]. These models are based on hard clustering, and the clustering preferences are expressed as *hard pairwise constraints* that *must* be satisfied. Some other authors [20, 3] extended their models to deal with soft pairwise constraints, where each constraint is assigned a weight. The performance of those constrained K-means algorithms is often not satisfactory, largely due to the incapability of K-means to model non-spherical data distribution in each class.

Shental et. al. [16] proposed a Gaussian mixture model (GMM) for clustering that incorporates hard pairwise constraints. However, the model cannot be naturally generalized to soft constraints, which are appropriate when our knowledge is only clustering preferences or carries significant uncertainty. Motivated in part to remedy this deficiency, Law et. al. [12, 13] proposed another GMM-based model to incorporate soft constraints. In their model, virtual groups are created for samples that are supposed to be in one class. The uncertainty information in pairwise relations is there expressed as the soft membership of samples to the virtual group. This modeling strategy is cumbersome to model samples shared by different virtual groups. Moreover, it cannot handle the prior knowledge that two samples are in different clusters. Other efforts to make use of the pairwise relations include changing the metric in feature space in favor of the specified relations [6, 22] or combining the metric learning with constrained clustering [4].

In this chapter, we describe a soft clustering algorithm based on GMM that expresses clustering preferences (in the form of pairwise relations) in the *prior probability on assignments of data points to clusters*. Our algorithm naturally accommodates both *hard constraints* and *soft preferences* in a framework in which the preferences are expressed as a Bayesian prior probability that pairs of points should (or should not) be assigned to the same cluster. After training with the Expectation-Maximization (EM) algorithm, the information expressed as a prior on the cluster assignment of the training data is successfully encoded in the means, covariances, and cluster priors in the GMM. Hence the model generalizes in a way consistent with the prior knowledge. We call the algorithm Penalized Probabilistic Clustering (PPC). Experiments on artificial and real-world data sets demonstrate that PPC can consistently improve the clustering result by incorporating reliable prior knowledge.

1.2 Model

Penalized Probabilistic Clustering (PPC) begins with a standard M-component GMM

$$P(x|\Theta) \;=\; \sum_{k=1}^{M} \pi_k \; P(x|\,\theta_k)$$

with the parameter vector $\Theta = \{\pi_1, \ldots, \pi_M, \theta_1, \ldots, \theta_M\}$. Here, π_k and θ_k are respectively the prior probability and parameters of the k^{th} Gaussian component. We augment the data set $X = \{x_i\},\ i = 1 \ldots N$ with (latent) cluster assignments $Z = \{z(x_i)\}, i = 1, \ldots, N$ to form the familiar *complete data* (X, Z). The complete data likelihood is

$$P(X, Z|\Theta) = P(X|Z, \Theta)P(Z|\Theta), \tag{1.1}$$

where $P(X|Z, \Theta)$ is the probability of X conditioned on Z

$$P(X|Z, \Theta) = \prod_{i=1}^{N} P(x_i|\theta_{z_i}). \tag{1.2}$$

1.2.1 Prior Distribution On Cluster Assignments

We incorporate our clustering preferences by manipulating the *prior probability* $P(Z|\Theta)$. In the standard Gaussian mixture model, the prior distribution on cluster assignments Z is trivial:

$$P(Z|\Theta) = \prod_{i=1}^{N} \pi_{z_i}. \tag{1.3}$$

We incorporate our clustering preferences through a weighting function $g(Z)$ that has large values when the assignment of data points to clusters Z conforms to our preferences, and low values when Z conflicts with our preferences. We can thus define the penalized prior as proportional to product of the original prior and the weighting factor:

$$P_p(Z|\Theta, G) \equiv \frac{(\prod_i \pi_{z_i})g(Z)}{\sum_Z (\prod_j \pi_{z_j})g(Z)} = \frac{1}{\Omega}(\prod_i \pi_{z_i})g(Z), \tag{1.4}$$

where $\Omega = \sum_Z (\prod_j \pi_{z_j})g(Z)$ is the normalization constant. Note that in equation (1.4), we use $P_p(\cdot)$ for the penalized prior, thus we can distinguish it from the standard one. This notation convention will be used throughout this chapter.

The likelihood of the data, *given a specific cluster assignment* Z, is independent of the cluster assignment preferences:

$$P(X, Z|\Theta, G) = P(X|Z, \Theta)P(Z|\Theta, G). \tag{1.5}$$

From equation (1.2), (1.4) and (1.5), the complete data likelihood is

$$P_p(X, Z|\Theta, G) = P(X|Z, \Theta)\frac{1}{\Omega}\prod_i \pi_{z_i} g(Z) = \frac{1}{\Omega}P(X, Z|\Theta)g(Z), \tag{1.6}$$

where $P(X, Z|\Theta)$ is the complete data likelihood for a *standard* GMM. The data likelihood is the sum of complete data likelihood over all possible Z, that is, $L(X|\Theta) = P_p(X|\Theta, G) = \sum_Z P_p(X, Z|\Theta, G)$, which can be maximized with the EM algorithm. Once the model parameters are fit, we do soft clustering according to the posterior probabilities for new data $P(k|x, \Theta)$. (Note that cluster assignment preferences are *not* expressed for the new data, only for the training data.)

1.2.2 Pairwise Relations

Pairwise relations provide a special case of the framework discussed above. The weighting factor given to the cluster assignment Z is:

$$g(Z) = \prod_{i \neq j} \exp(w(i, j)\ \delta_{z_i z_j}), \tag{1.7}$$

where $w(i, j)$ is the weight associated with sample pair (x_i, x_j), with

$$w(i, j) \in [-\infty, \infty],\ w(i, j) = w(j, i).$$

The weight $w(i, j)$ reflects our preference for assigning x_i and x_j into one cluster: We use $w(i, j) > 0$ if $(i, j) \in C_=$, $w(i, j) < 0$ when $(i, j) \in C_{\neq}$, and $w(i, j) = 0$ if no constraint is specified for x_i and x_j. The absolute value $|w(i, j)|$ reflects the strength of the preference. The prior probability with the pairwise relations is

$$P(Z|\Theta, G) = \frac{1}{\Omega}\prod_i \pi_{z_i} \prod_{i \neq j} \exp(w(i, j)\ \delta_{z_i z_j}). \tag{1.8}$$

From equation (1.7) and (1.8), the weighting factor $g(Z)$ is large when the pairwise constraints expressed through $W = \{w(i, j)\}$ are satisfied by the cluster assignment Z.

The model described above provides a fairly flexible framework that encompasses standard GMM and several other constrained clustering models as special cases. Most obviously, when we let $w(i, j) = 0$ for all i and j, we have $g(Z) = 1$ for all Z, hence the complete likelihood reduces to the standard one:

$$P_p(X, Z|\Theta, G) = \frac{1}{\Omega}P(X, Z|\Theta)g(Z) = P(X, Z|\Theta). \tag{1.9}$$

In the other extreme with $|w(i,j)| \to \infty$, assignments Z that violate constraint between x_i and x_j have zero prior probability, since for those assignments

$$P_p(Z|\Theta, G) = \frac{\prod_k \pi_{z_k} \prod_{i \neq j} \exp(w(i,j)\ \delta_{z_i z_j})}{\sum_Z \prod_l \pi_{z_l} \prod_{m \neq n} \exp(w(m,n)\ \delta_{z_m z_n})} \to 0.$$

Then the relations become *hard constraints*, while the relations with $|w(i,j)| < \infty$ are called *soft preferences*. When all the specified pairwise relations are hard constraints, the data likelihood becomes

$$P_p(X, Z|\Theta, G) = \frac{1}{\Omega} \prod_{ij \in C_=} \delta_{z_i z_j} \prod_{ij \in C_{\neq}} (1 - \delta_{z_i z_j}) \prod_{i=1}^{N} \pi_{z_i} P(x_i|\, \theta_{z_i}). \quad (1.10)$$

It is straightforward to verify that equation (1.10) is essentially the same with the complete data likelihood given by [16]. In Appendix A, we give a detailed derivation of equation (1.10) and hence the equivalence of two models. When only hard constraints are available, we simply implement PPC based on equation (1.10). In the remainder of this chapter, we will use W to denote the prior knowledge on pairwise relations, that is

$$P_p(X, Z|\Theta, G) \equiv P_p(X, Z|\Theta, W) = \frac{1}{\Omega} P(X, Z|\Theta) \prod_{i \neq j} \exp(w(i,j)\ \delta_{z_i z_j}) \quad (1.11)$$

1.2.3 Model Fitting

We use the EM algorithm [7] to fit the model parameters Θ:

$$\Theta^* = \arg\max_{\Theta} L(X|\Theta, W)$$

The expectation step (E-step) and maximization step (M-step) are

$$\texttt{E-step: } Q(\Theta, \Theta^{(t-1)}) = E_{Z|X}(\log P_p(X, Z|\Theta, W)|X, \Theta^{(t-1)}, W)$$
$$\texttt{M-step: } \Theta^{(t)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(t-1)}).$$

In the M-step, the optimal mean and covariance matrix of each component is:

$$\mu_k = \frac{\sum_{j=1}^{N} x_j P_p(k|x_j, \Theta^{(t-1)}, W)}{\sum_{j=1}^{N} P_p(k|x_j, \Theta^{(t-1)}, W)}$$
$$\Sigma_k = \frac{\sum_{j=1}^{N} P_p(k|x_j, \Theta^{(t-1)}, W)(x_j - \mu_k)(x_j - \mu_k)^T}{\sum_{j=1}^{N} P_p(k|x_j, \Theta^{(t-1)}, W)}\ .$$

The update of the prior probability of each component is more difficult due to the normalizing constant Ω in the data likelihood

$$\Omega = \sum_Z \{\prod_{k=1}^{N} \pi_{z_k} \prod_{i \neq j} \exp(w(i,j)\ \delta_{z_i z_j})\}. \quad (1.12)$$

We need to find

$$\pi \equiv \{\pi_1, \ldots, \pi_m\} = \arg\max_{\pi} \sum_{l=1}^{M} \sum_{i=1}^{N} \log \pi_l P_p(l|x_i, \Theta^{(t-1)}, W) - \log \Omega(\pi), \tag{1.13}$$

which, unfortunately, does not have a closed-form solution in general [1]. In this chapter, we use a rather crude approximation of the optimal π instead. First, we estimate the values of $\log \Omega(\pi)$ on a grid $H = \{\hat{\pi}^n\}$ on the simplex defined by

$$\sum_{k=1}^{M} \pi_k = 1,\ \pi_k \geq 0.$$

Then in each M-step, we calculate the value of $\sum_{l=1}^{M} \sum_{i=1}^{N} \log \hat{\pi}_l^n P_p(l|x_i, \Theta^{(t-1)}, W)$ for each node $\hat{\pi}^n \in H$ and find the node $\hat{\pi}^*$ that maximizes the function defined in equation (1.13):

$$\hat{\pi}^* = \arg\max_{\hat{\pi}^n \in H} \sum_{l=1}^{M} \sum_{i=1}^{N} \log \hat{\pi}_l^n P_p(l|x_i, \Theta^{(t-1)}, W) - \log \Omega(\hat{\pi}^n). \tag{1.14}$$

We use $\hat{\pi}^*$ as the approximative solution of equation (1.13). In this chapter, the resolution of the grid is set to be 0.01. Although it works very well for all experiments in this chapter, we notice that the search over grid will be fairly slow for $M > 5$. [17] proposed to find optimal π using gradient descent and approximate $\Omega(\pi)$ by pretending all specified relations are disjoint (see §1.3.1). Although this method is originally designed for hard constraints, it can be easily adapted for PPC. This will not be covered in this chapter.

It is important to note that with a non-trivial w, the cluster assignment of samples are no longer independent to each other, consequently the posterior estimation of each sample can not be done separately. This fact brings extra computational problem and will be discussed later in §1.3.

1.2.4 Selecting the Constraint Weights

1.2.4.1 Example: How the Weight w Affects Clustering

The weight matrix W is crucial to the performance of the PPC. Here we give an example demonstrating how the weight of pairwise relations affects the clustering process. Figure 1.1 (a) shows the 2-dimensional data sampled from four spherical Guassians centered at (-1,-1), (-1,1), (1,-1), and (1,1). We intend to group the data into two classes, as indicated by the symbols. Besides the data set, we also have 20 pairs correctly labeled as must-links and

[1] [16] pointed out that with a different sampling assumption, a closed-form solution for equation (1.13) exists when only hard must-links are available. See §1.4.

cannot-links, as shown in Figure 1.1 (b). We try to fit the data set with a two-component GMM. Figure 1.1 (c) and (d) give the density contour of the two possible models on the data. Without any pairwise relations specified, we have approximately equal chance to get each GMM model. After incorporating pairwise relations, the EM optimization process is biased towards the intended one. The weights of pairwise relations are given as follows

$$w(i,j) = \begin{cases} w & (x_i, x_j) \in C_= \\ -w & (x_i, x_j) \in C_{\neq} \\ 0 & \text{otherwise,} \end{cases}$$

where $w \geq 0$ measures the certainty of all specified pairwise constraints. In Figure 1.2, we give three runs with *same* initial model parameters but different weight for constraints.

For each run, we give snapshots of model after 1, 3, 5 and 20 EM iterations. The first row is the run with $w = 0$ (standard GMM). The search ends up with a model that violates our prior knowledge of class membership. The middle row is the run with w set to 1.3, with the same poor initial condition, the model fitting process still goes to the wrong one again, although at a slower pace. In the bottom row, we increase w to 3, this time the model converges to the one we intend.

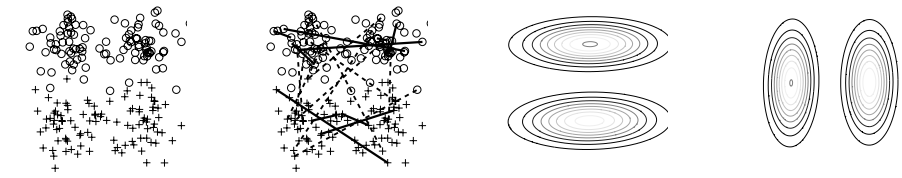

FIGURE 1.1: The influence of constraint weight on model fitting. (a) artificial data set. (b) must-links (solid lines) and cannot-links (dotted line). (c) and (d): the probability density contour of two possible fitted models.

1.2.4.2 Choosing Weight w Based on Prior Knowledge

There are some occasions we can translate our prior belief on the relations into the weight W. Here we assume that the pairwise relations are labeled by an oracle but contaminated by flipping noise before they are delivered to us. For each labeled pair (x_i, x_j), there is thus a certainty value $0.5 \leq \gamma_{ij} \leq 1$

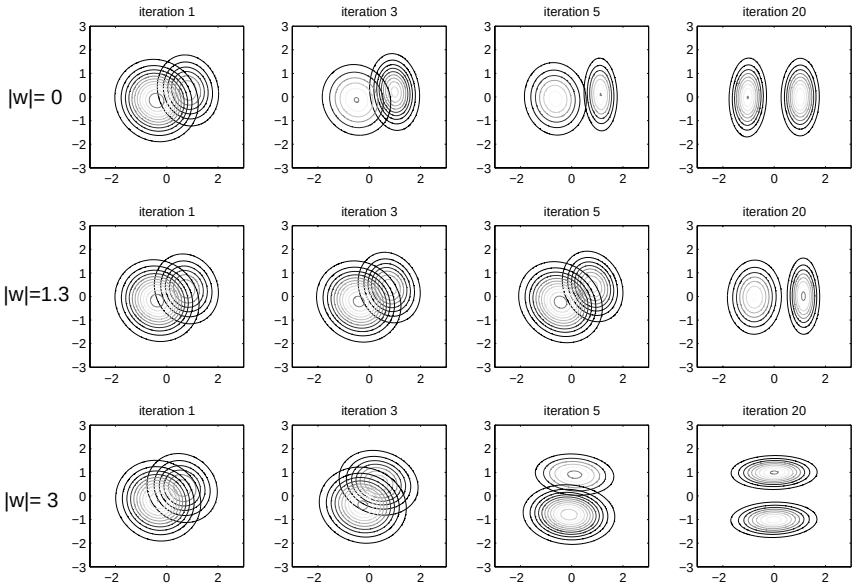

FIGURE 1.2: The contour of probability density fit on data with different weight given to pairwise relations. Top row: $w = 0$; Middle row: w=1.3; Bottom row: $w = 3$.

equal to the probability that pairwise relation is *not* flipped [2]. Our prior knowledge would include those specified pairwise relations and their certainty values $\Gamma = \{\gamma_{ij}\}$.

This prior knowledge can be *approximately* encoded into the weight w by letting

$$w(i,j) = \begin{cases} \frac{1}{2}\log(\frac{\gamma_{ij}}{1-\gamma_{ij}}) & (x_i, x_j) \text{ is specified as must-linked} \\ -\frac{1}{2}\log(\frac{\gamma_{ij}}{1-\gamma_{ij}}) & (x_i, x_j) \text{ is specified as cannot-linked} \\ 0 & \text{otherwise.} \end{cases} \quad (1.15)$$

The details of the derivation are in Appendix B. It is obvious from equation (1.15) that for a specified pairwise relation (x_i, x_j), the greater the certainty value γ_{ij}, the greater the absolute value of weight $w(i,j)$.

Note that the weight designed this way is not necessarily optimal in terms of classification accuracy, as will be demonstrated by experiment in §1.5.1. The reason is twofold. First, equation (1.15) is derived based on a (possibly crude) approximation. Second, Gaussian mixture models, as classifiers, are often considerably biased from true class distribution of data. As a result, even

[2]We only consider the certainty value > 0.5, because a pairwise relation with certainty $\gamma_{ij} < 0.5$ can be equivalently treated as its opposite relation with certainty $1 - \gamma_{ij}$.

if the PPC prior $P(Z|\Theta, W)$ faithfully reflects the truth, it does not necessarily lead to the best classification accuracy. Nevertheless, equation (1.15) gives a good initial guidance for choosing the weight. Our experiments in §1.5.1 show that this design often yields superior classification accuracy than simply using the hard constraints or ignoring the pairwise relations (standard GMM).

This weight design scheme is directly applicable when pairwise relations are labeled by domain experts and the certainty values are given at the same time. We might also *estimate* the flipping noise parameters from historical data or from available statistics. For example, we can derive soft pairwise relations based on spatial or temporal continuity among samples. That is, we add soft must-links to all adjacent pair of samples, assuming the flipping noise explaining all the adjacent pairs that are actually *not* in one class. We further assume that the flipping noise each pair follows the same distribution. Accordingly we assign the same weight to all adjacent pairs. Let q denote the probability that the label on a adjacent pair is flipped. We might be able to estimate q from labeled instances of a similar problem, for example, segmented images or time series. The maximum likelihood (ML) estimation of q is given by simple statistics:

$$\tilde{q} = \frac{\text{the number of adjacent pairs that are not in the same class}}{\text{the number of all adjacent pairs}}.$$

We give an application of this idea in §1.5.2.

1.3 Computing the Cluster Posterior

Both the M-step and the final clustering require the cluster membership posterior. Computing this posterior is simple for the standard GMM since each data point x_i is assigned to a cluster independently. The pairwise constraints bring extra relevancy in assignment among samples involved. From equation (1.11), if $w(i, j) \neq 0$,

$$P_p(z_i, z_j|x_i, x_j, \Theta, W) \neq P_p(z_i|x_i, \Theta, W)P_p(z_j|x_j, \Theta, W). \tag{1.16}$$

This relevancy can be further extended to any two pair x_i and x_j that are connected by a path of nonzero weights. Clearly $P_p(X, Z|\theta, W)$ can be best described as a undirected graphical model, and the exact inference of the posterior must be based on the maximal connected subgraphs (MCS). The inference of sample x_i in a MCS T based on a brutal force marginalization is

$$P_p(z_i = k|X, \Theta, W) = \sum_{Z_T|z_i=k} P_p(Z_T|X_T, \Theta, W),$$

which requires time complexity $O(M^{|T|})$. This calculation can get prohibitively expensive if $|T|$ is very big.

We will firs give some special cases with easy inference, then we will discuss Gibbs sampling and mean field approximation as two approximate inference models used for estimating the posterior. Other methods for complicated graphical model inference, such as (loopy) belief propagation, are also proposed to solve this kind of problems [8].

1.3.1 Two Special Cases with Easy Inference

Apparently the inference is easy when we limit ourselves to small MCS. Specifically, when $|T| \leq 2$, the pairwise relations are *disjoint*. With disjoint constraints, the posterior probability for the whole data set can be given in closed-form with $O(N)$ time complexity. Moreover, the evaluation of the normalization factor $\Omega(\pi)$ is simple:

$$\Omega(\pi) = (1 - \sum_{k=1}^{M} \pi_k^2)^{|C_=|} (\sum_{k=1}^{M} \pi_k^2)^{|C_\neq|}.$$

The optimization of π in M-step can thus be achieved with little cost. Sometimes disjoint relations are a natural choice: they can be generated by picking up sample pairs from sample set and labeling the relations *without replacement*. More generally, we can avoid the expensive computation in posterior inference by breaking large MCS into small ones. To do this, we need to deliberately ignore some pairwise constraints. In §1.5.2, Experiment 2 is an application of this idea.

The second simplifying situation is when we have only hard must-links ($w(i,j) = +\infty$ or 0). Since must-link is an equivalence relation, we group the data set into several equivalence classes (called chunklets). Each chunklet can be treated as a single sample. That is, assume x_i is in chunklet T, we then have

$$P_p(z_i = k|x_i, \Theta, W) = P_p(Z_T = k|x_T, \Theta, W) = \frac{\prod_{j \in T} \pi_k P(x_j|\theta_k)}{\sum_{k'} (\prod_{j \in T} \pi_{k'} P(x_j|\theta_{k'}))}.$$

Similar ideas have been proposed independently in [21, 16, 4]. This case is useful when we are sure that a group of samples are from one source [16].

For more general cases where the exact inference is computationally prohibitive, we propose to use Gibbs sampling [15] and the mean field approximation [9] to estimate the posterior probability. This will be discussed in §1.3.2 and §1.3.3.

1.3.2 Estimation with Gibbs Sampling

In Gibbs sampling, we estimate $P_p(z_i|X, \Theta, W)$ as a sample mean

$$P_p(z_i = k|X, \Theta, W) = E(\delta_{z_i k}|X, \Theta, W) \approx \frac{1}{S} \sum_{t=1}^{S} \delta_{z_i^{(t)} k},$$

where the sum is over a sequence of S samples from $P(Z|X,\Theta,G)$ generated by the Gibbs MCMC. The t^{th} sample in the sequence is generated by the usual Gibbs sampling technique:

- Pick $z_1^{(t)}$ from distribution $P_p(z_1|z_2^{(t-1)}, z_3^{(t-1)}, ..., z_N^{(t-1)}, X, w, \Theta)$
- Pick $z_2^{(t)}$ from distribution $P_p(z_2|z_1^{(t)}, z_3^{(t-1)}, ..., z_N^{(t-1)}, X, w, \Theta)$
  ...
- Pick $z_N^{(t)}$ from distribution $P_p(z_N|z_1^{(t)}, z_2^{(t)}, ..., z_{N-1}^{(t)}, X, w, \Theta)$

For pairwise relations it is helpful to introduce some notation. Let Z_{-i} denote an assignment of data points to clusters that leaves out the assignment of x_i. Let $U(i)$ be the indices of the set of samples that participate in a pairwise relation with sample x_i, $U(i) = \{j : w(i,j) \neq 0\}$. Then we have

$$P_p(z_i|Z_{-i}, X, \Theta, W) \propto P(x_i, z_i|\Theta) \prod_{j \in U(i)} \exp(2w(i,j)\ \delta_{z_i z_j}). \tag{1.17}$$

The time complexity of each Gibbs sampling pass is $O(NnM)$, where n is the maximum number of pairwise relations a sample can be involved in. When W is sparse, the size of $U(i)$ is small, thus calculating $P_p(z_i|Z_{-i}, X, \Theta, W)$ is fairly cheap and Gibbs sampling can effectively estimate the posterior probability.

1.3.3 Estimation with Mean Field Approximation

Another approach to posterior estimation is to use mean field theory [9, 11]. Instead of directly evaluating the intractable $P_p(Z|X,\Theta,W)$, we try to find a tractable mean field approximation $Q(Z)$. To find a $Q(Z)$ close to the true posterior probability $P_p(Z|X,\Theta,W)$, we minimize the Kullback-Leibler divergence between them, i.e.

$$\min_Q \mathrm{KL}(Q(Z)|P_p(Z|X,\Theta,W)), \tag{1.18}$$

which can be recasted into:

$$\max_Q [H(Q) + E_Q\{\log P_p(Z|X,\Theta,W)\}], \tag{1.19}$$

where $E_Q\{\cdot\}$ denotes the expectation with respect to Q. The simplest family of variational distribution is one where all the latent variables $\{z_i\}$ are independent of each other:

$$Q(Z) = \prod_{i=1}^{N} Q_i(z_i). \tag{1.20}$$

With this $Q(Z)$, the optimization problem in equation (1.19) does not have a closed-form solution, nor is it a convex problem. Instead, a locally optimal Q can be found iteratively with the following update equations

$$Q_i(z_i) \leftarrow \frac{1}{\Omega_i} \exp(E_Q\{\log P_p(Z|X,\Theta,W)|z_i\}) \tag{1.21}$$

for all i and $z_i \in \{1, 2, \cdots, M\}$. Here $\Omega_i = \sum_{z_i} \exp(E_Q\{\log P_p(Z|X,\Theta,W)|z_i\})$ is the local normalization constant. For the PPC model, we have

$$\exp(E_Q\{\log P_p(Z|X,\Theta,W)|z_i\}) = P(z_i|x_i,\Theta) \exp(\sum_{j\neq i} w(i,j)Q_j(z_i)).$$

Equation (1.21), collectively for all i, are the *mean field equations*. Evaluation of mean field equations requires at most $O(NnM)$ time complexity, which is same as the time complexity of one Gibbs sampling pass. Successive updates of equation (1.21) will converge to a local optimum of equation (1.19). In our experiments, the convergence usually occurs after about 20 iterations, which is much less than the number of passes required for Gibbs sampling.

1.4 Related Models

Prior to our work, different authors have proposed several constrained clustering models based on K-means, including the seminal work by Wagstaff and colleagues [21, 20], and its successor [2, 3, 4]. These models generally fall into two classes. The first class of algorithms [21, 2] keep the original K-means cost function (reconstruction error) but confine the cluster assignments to be consistent with the specified pairwise relations. The problem can be casted into the following constrained optimization problem

$$\begin{aligned} \min_{Z,\mu} \quad & \sum_{i=1}^{N} ||x_i - \mu_{z_i}||^2 \\ \text{subject to} \quad & z_i = z_j, \text{ if } (x_i, x_j) \in C_= \\ & z_i \neq z_j, \text{ if } (x_i, x_j) \in C_{\neq}, \end{aligned}$$

where $\mu = \{\mu_1, \cdots, \mu_M\}$ is the cluster centers. In the second class of algorithms, cluster assignments that violate the pairwise relations are allowed, but will be penalized. They employ a modified cost function [3]:

$$J(\mu, Z) = \frac{1}{2}\sum_{i=1}^{N} ||x_i - \mu_{z_i}||^2 + \sum_{(i,j)\in C_=} a_{ij}(z_i \neq z_j) + \sum_{(i,j)\in C_{\neq}} b_{ij}(z_i = z_j), \tag{1.22}$$

where a_{ij} is the penalty for violating the must-link between (x_i, x_j) and b_{ij} is the penalty when the violated pairwise relation is a cannot-link. It can be shown that both classes of algorithms are special cases of PPC with spherical Gaussian components and proper setting of radius and w (see Appendix C).

There are two weaknesses shared by the constrained K-means model. The first is their limited modeling capability inherited from the standard K-means. This weakness can be alleviated with the extra information from the pairwise constraints [20], but it often takes a lot pairwise constraints to really achieve decent results when the distribution of class can not be naturally modeled by K-means. As the second weakness, the hard clustering nature of constrained K-means often requires a combinatorial optimization of the cluster assignments, which is usually not trivial and often intractable. To cope with that, various ways have been proposed to obtain a suboptimal solution [2, 3, 21].

To overcome the limitation of constrained K-means, several authors proposed probabilistic constrained clustering models based on Gaussian mixture. The models proposed by Shental et. al. [16, 17] address the situation where pairwise relations are hard constraints. The authors partition the whole data set into a number of chunklets consisting of samples that are (hard) must-linked to each other[3]. They discuss two sampling assumptions:

- Assumption 1: chunklet X_i is generated i.i.d from component k with prior π_k [17], and the complete data likelihood is

$$P(X, Y|\Theta, E_\Omega) = \frac{1}{\Omega} \prod_{ij \in C_{\neq}} (1 - \delta_{z_i z_j}) \cdot \prod_{l=1}^{L} \{\pi_{z_l} \prod_{x_i \in X_l} P(x_i|\, \theta_{z_l})\}, \quad (1.23)$$

  where E_Ω denotes the specified constraints.

- Assumption 2: chunklet X_i generated from component k with prior $\propto \pi_k^{|X_i|}$, where $|X_i|$ is the number of samples in X_i [17]. The complete data likelihood is:

$$P(X, Y|\Theta, E_\Omega) = \frac{1}{\Omega} \prod_{ij \in C_{\neq}} (1 - \delta_{z_i z_j}) \cdot \prod_{l=1}^{L} \{\pi_{z_l}^{|X_l|} \prod_{x_i \in X_l} P(x_i|\, \theta_{z_l})\} \quad (1.24)$$

$$= \frac{1}{\Omega} \prod_{ij \in C_{=}} \delta_{z_i z_j} \prod_{ij \in C_{\neq}} (1 - \delta_{z_i z_j}) \prod_{i=1}^{N} \pi_{z_i} P(x_i|\, \theta_{z_i}). \quad (1.25)$$

In Appendix A we show that when using Assumption 2, the model expressed in equation (1.24)-(1.25) is equivalent to PPC with only hard constraints (as expressed in equation (1.10)). It is suggested in [17] that Assumption 1 might be appropriate, for example, when chunklets are generated from temporal

[3] If a sample is not must-linked to any other samples, it comprises a chunklet by itself.

continuity. When pairwise relations are generated by labeling sample pairs picked from data set, Assumption 2 might be more reasonable. Assumption 1 allows a closed-form solution in the M-step, including solution for π, in each EM iteration [17].

To incorporate the uncertainty associated with pairwise relations, Law et. al. [12, 13] proposed to use soft group constraints. To model a must-link between any sample pair (x_i, x_j), they create a group l and express the strength of the must-link as the membership of x_i and x_j to group l. This strategy works well for some simple situations, for example, when the pairwise relations are disjoint(as defined in §1.3.1). However, it is awkward if samples are shared by multiple groups, which is unavoidable when samples are commonly involved in multiple relations. Another serious drawback of the group constraints model is its inability to model cannot-links. Due to these obvious limitations, we omit the empirical comparison of this model to PPC in the following experiment section.

1.5 Experiments

The experiments section consists of two parts. In §1.5.1, we empirically evaluate the influence of randomly generated constraints on the clustering result when using PPC, and compare it with other constrained clustering algorithms. In §1.5.2, we address real-world problems, where the constraints are derived from our prior knowledge. Also in this section, we demonstrate the approaches to reduce computational complexity, as described in §1.3.

Following are some abbreviations we will use throughout this section: *soft-PPC* for PPC with soft constraints, *hard-PPC* for PPC with hard constraints (implemented based on equation (1.10)), *soft-CKmeans* for the K-means with soft constraints [3] and *hard-CKmeans* for the K-means with hard constraints [21]. The Gaussian mixture model with hard constraints [16, 17] will be referred to as constrained-EM.

1.5.1 Artificial Constraints

In this section, we discuss the influence of pairwise relations on PPC's clustering. Due to the equivalence between hard-PPC and constrained-EM algorithm [17], we will not repeat the experiments with correct constraints and hard constraints in Chapter ???. Instead, we consider the more general situation where pairwise constraints are noisy, and thus justify the use of soft-PPC. The weights of soft-PPC are designed based on the strategy described in §1.2.4. The result is compared to hard-PPC and other semi-supervised clustering models.

Constraint Selection: We chose to limit our discussion to the disjoint pairwise relations, and leave to the more complicated cases to §1.5.2. As discussed in §1.3.1, the disjoint pairwise relations, hard or soft, allows fast solution in the maximization step in each EM iteration. The pairwise relations are generated as follows: we randomly pick two samples from the data set without replacement. If the two have the same class label, we then add a must-link constraint between them; otherwise, we add a cannot-link constraint. After the constraints are chosen, we add a noise to all the constraints by randomly flipping each pairwise relation with a certain probability $q \leq 0.5$. For the soft-PPC model, the weight $w(i,j)$ to each specified pairwise relation is given as follows:

$$w(i,j) = \begin{cases} \frac{1}{2}\log(\frac{1-q}{q}) & (x_i, x_j) \text{ specified as must-link} \\ -\frac{1}{2}\log(\frac{1-q}{q}) & (x_i, x_j) \text{ specified as cannot-link.} \end{cases} \tag{1.26}$$

For soft-CKmeans, we give equal weights to all the specified constraints. Because there is no guiding rule in literature on how to choose weight for soft-CKmeans model, we simply use the weight that yields the highest classification accuracy.

Performance Evaluation: We try PPC (with the number of components equal to the number of classes) with various numbers of pairwise relations. For comparison, we also give results of standard GMM, standard K-means, hard-CKmeans [21], and hard-PPC. For each clustering result, a confusion matrix is built to compare it to true labeling. The classification accuracy is calculated as the ratio of the sum of diagonal elements to the number of all samples. The reported classification accuracy is averaged over 100 different realizations of pairwise relations.

Experiment 1: Artificial Constraints

In this experiment, we evaluate the performance of soft-PPC with noisy constraints on three two-dimensional artificial data sets and three UCI data sets. The three two-dimensional artificial data sets (Figure 1.3) are designed to highlight PPC's superior modeling flexibility over constrained K-means[4]. In each example, there are 200 samples in each class. We perform the same experiments on three UCI data sets: the Iris data set has 150 samples and three classes, 50 samples in each class; the Waveform data set has 5000 samples and three classes, around 1700 samples in each class; the Pendigits data set includes four classes (digits 0, 6, 8, 9), each with 750 samples.

On each data set, we randomly generate a number of disjoint pairwise relations to have 50% of the data involved. In this experiment, we try two

[4]Some authors [22, 6, 4] combined standard or constrained K-means with metric learning based on pairwise relations, and reported improvement on classification accuracy. This will not be discussed in this chapter.

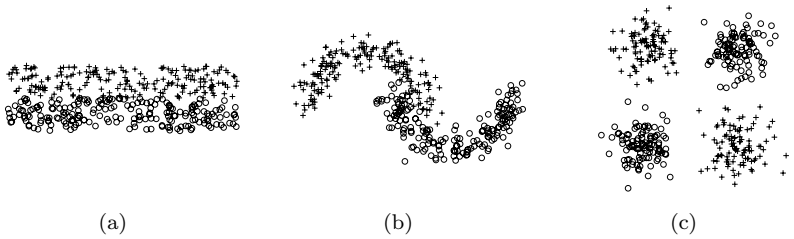

FIGURE 1.3: Three artificial data sets, with class denoted by symbols.

different noise levels with q set to 0.15 and 0.3. Figure 1.4 compares the classification accuracies given by the maximum likelihood (ML) solutions[5] of different models. The accuracy for each model is averaged over 20 random realizations of pairwise relations. On all data sets except artificial data set 3, soft-PPC with the designed weight gives higher accuracy than hard-PPC and standard GMM on both noise levels. On artificial data set 3 , when $q = 0.3$ hard-PPC gives the best classification accuracy [6]. Soft-PPC apparently gives superior classification accuracy to the K-means models on all six data sets, even though the weight of soft-CKmeans is optimized. Figure 1.4 also shows that it can be harmful to use hard constraints when pairwise relations are noisy, especially when the noise is significant. Indeed, as shown by Figure 1.4 (d) and (f), hard-PPC can yield accuracy even worse than standard GMM.

1.5.2 Real World Problems

In this section, we present two examples where pairwise constraints are from domain experts or common sense. Both examples are about image segmentation based on Gaussian mixture models. In the first problem (Experiment 2), hard pairwise relations are derived from image labeling done by a domain expert. In the second problem, soft pairwise relations are generated based on spatial continuity.

Experiment 2: Hard Do-not-links from Partial Class Information

The experiment in this subsection shows the application of pairwise constraints on partial class information. For example, consider a problem with six classes $A, B, ..., F$. The classes are grouped into several class-sets $C_1 = \{A, B, C\}, C_2 = \{D, E\}, C_3 = \{F\}$. The samples are partially labeled in the

[5] We choose the one with the highest data likelihood among 100 runs with different random initialization. For K-means models, including soft-CKmeans and hard-CKmeans, we use the solutions with the smallest cost.

[6] Further experiment shows that on this data, soft-PPC with the optimal w ($>$ the one suggested by equation (1.26)) is still slightly better than hard-PPC.

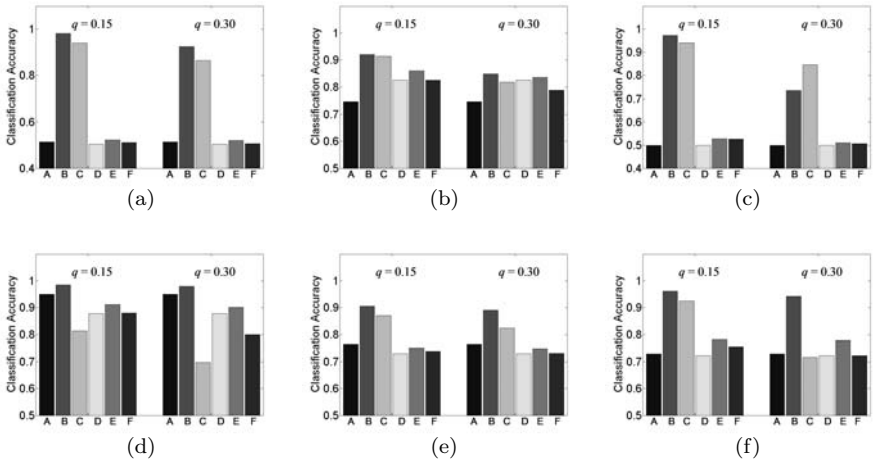

FIGURE 1.4: Classification accuracy with noisy pairwise relations. We use all the data in clustering. In each panel, **A**: standard GMM; **B**: soft-PPC; **C**: hard-PPC; **D**: standard K-means; **E**: soft-CKmeans with optimal weight; **F**: hard-CKmeans.

sense that we are told which class-set a sample is from, but not which specific class it is from. We can logically derive a cannot-link constraint between any pair of samples known to belong to different class-sets, while no must-link constraint can be derived if each class-set has more than one class in it.

Figure 1.5 (a) is a 120x400 region from Greenland ice sheet from NASA Langley DAAC [7] [18]. Each pixel has intensities from seven spectrum bands. This region is labeled into snow area and non-snow area, as indicated in Figure 1.5 (b). The snow area may contain samples from several classes of interest: ice, melting snow and dry snow, while the non-snow area can be bare land, water or cloud. The labeling from expert contains incomplete but useful information for further segmentation of the image. To segment the image, we first divide it into 5x5x7 blocks (175 dim vectors). We use the first 50 principal components as feature vectors. Our goal is then to segment the image into (typically > 2) areas by clustering those feature vectors. With PPC, we can encode the partial class information into do-not-link s.

For hard-PPC, we use half of the data samples for training, and the rest for test. Hard cannot-link constraints (only on training set) are generated as

[7]We use the first seven MoDerate Resolution Imaging Spectroradiometer (MODIS) Channels with bandwidths as follows (in nm): Channel 1: 620-670, Channel 2: 841-876, Channel 3: 459-479, Channel 4: 545-565, Channel 5: 1230-1250, Channel 6: 1628-1652, Channel 7:2105-2155

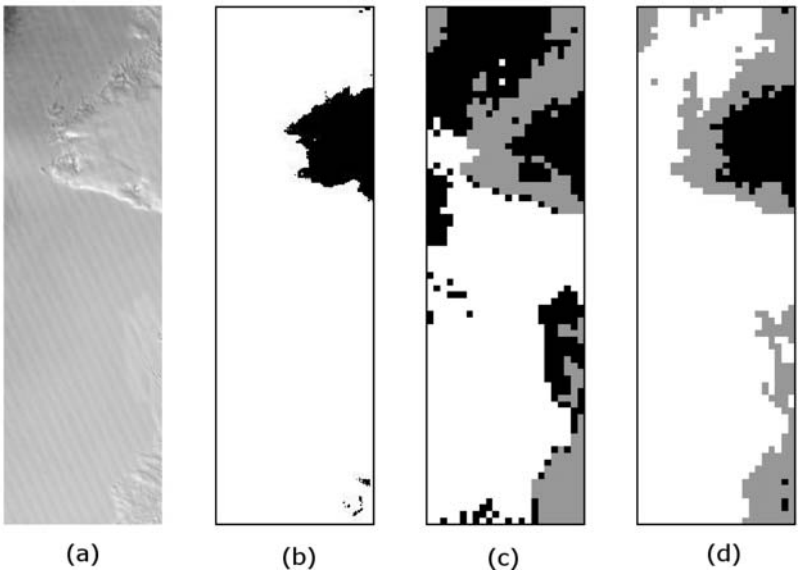

FIGURE 1.5: (a) Gray-scale image from the first spectral channel 1. (b) Partial label given by expert, black pixels denote non-snow area and white pixels denote snow area. Clustering result of standard GMM (c) and PPC (d). (c) and (d) are colored according to image blocks' assignment.

follows: for each block in the non-snow area, we randomly choose (without replacement) six blocks from the snow area to build cannot-link constraints. By doing this, we achieve cliques with size seven (1 non-snow block + 6 snow blocks). As in §1.5.1, we apply the model fit with hard-PPC to the test set and combine the clustering results on both data sets into a complete picture. Clearly, the clustering task is non-trivial for any $M > 2$. A typical clustering result of 3-component standard GMM and 3-component PPC are shown as Figure 1.5 (c) and (d) respectively. Standard GMM gives a clustering that is clearly in disagreement with the human labeling in Figure 1.5 (b). The hard-PPC segmentation makes far fewer mis-assignments of snow areas (tagged white and gray) to non-snow (black) than does the GMM. The hard-PPC segmentation properly labels almost all of the non-snow regions as non-snow. Furthermore, the segmentation of the snow areas into the two classes (not labeled) tagged white and gray in Figure 1.5 (d) reflects subtle differences in the snow regions captured by the gray-scale image from spectral channel 1, as shown in Figure 1.5 (a).

Experiment 3: Soft Links from Continuity

In this subsection, we will present an example where soft constraints come from continuity. As in the previous experiment, we try to do image segmen-

tation based on clustering. The image is divided into blocks and rearranged into feature vectors. We use a GMM to model those feature vectors, with each Gaussian component representing one texture. However, standard GMM often fails to give good segmentations because it cannot make use of the spatial continuity of image, which is essential in many image segmentation models, such as random field [5]. In our algorithm, the spatial continuity is incorporated as the soft must-link preferences with uniform weight between each block and its neighbors. As described in §1.2.4, the weight w of the soft must-link can be given as

$$w = \frac{1}{2}\log(\frac{1-q}{q}), \tag{1.27}$$

where q is the ratio of softly-linked adjacent pairs that are not in the same class. Usually q is given by an expert or estimated from segmentation result of similar images. In this experiment, we assume we already know the ratio q, which is calculated from the label of the image.

The *complete* data likelihood is

$$P_p(X, Z|\Theta, W) = \frac{1}{\Omega}P(X, Z|\Theta)\prod_i \prod_{j\in U(i)} \exp(w\ \delta_{z_i z_j}), \tag{1.28}$$

where $U(i)$ means the neighbors of the i^{th} block. The EM algorithm can be roughly interpreted as iterating on two steps: (1) estimating the texture description (parameters of mixture model) based on segmentation, and (2) segmenting the image based on the texture description given by step 1. Since exact calculation of the posterior probability is intractable due to the large clique containing all samples, we have to resort to approximation methods. In this experiment, both the Gibbs sampling (see §1.3.2) and the mean field approximation (see §1.3.3) are used for posterior estimation. For Gibbs sampling, equation (1.17) is reduced to

$$P_p(z_i|Z_{-i}, X, \Theta, W) \propto P(x_i, z_i|\Theta) \prod_{j\in U(i)} \exp(2w\ \delta_{z_i z_j}).$$

The mean field equation (1.21) is reduced to

$$Q_i(z_i) \leftarrow \frac{1}{\Omega_i}P(x_i, z_i|\Theta) \prod_{j\in U(i)} \exp(2w\ Q_j(z_i)).$$

The image shown in Figure 1.6 (a) is built from four Brodatz textures [8] . This image is divided into 7x7 blocks and then rearranged to 49-dim vectors.

[8]Downloaded from http://sipi.usc.edu/services/database/Database.html, April, 2004.

We use those vectors' first five principal components as the associated feature vectors. A typical clustering result of 4-component standard GMM is shown in Figure 1.6 (b). For soft-PPC, the soft must-links with weight w calculated from equation (1.27) are added between each block and its four neighbors. Figure 1.6 (c) and (d) are the clustering result of 4-component soft-PPC with respectively Gibbs sampling and mean field approximation. One run with Gibbs sampling takes around 160 minutes on a PC with Pentium 4, 2.0 G HZ processor whereas the algorithm using the mean field approximation takes only 3.1 minutes. Although mean field approximation is about 50 times faster than Gibbs sampling, the clustering result are comparable according to Figure 1.6. Comparing to the result given by standard GMM, soft-PPC with both approximation methods achieves significantly better segmentation after incorporating spatial continuity.

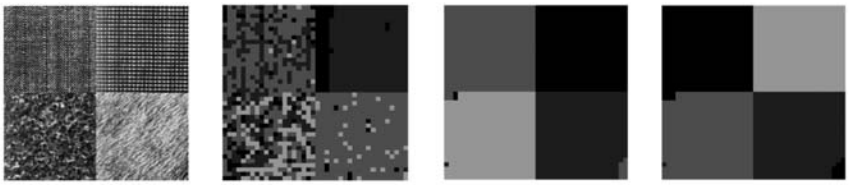

FIGURE 1.6: (a) Texture combination. (b) Clustering result of standard GMM. (c) Clustering result of soft-PPC with Gibbs sampling. (d) Clustering result of soft-PPC with mean field approximation. (b)-(d) are shaded according to the blocks assignments to clusters.

1.6 Discussion

Despite its success shown above, PPC has its limitations. First, PPC often needs a substantial *proportion* of samples involved in pairwise relations to give good results. Indeed, if we have the number of relations fixed and keep adding samples without any new relations, the algorithm will finally degenerate into unsupervised learning (clustering). To overcome this, one can instead build semi-supervised model based on discriminative models such as support vector machine (see Chapter ???) or Gaussian process classifier [14], and use the pairwise relations as observation. Second, since PPC is based on the Gaussian mixture model, it works well in the situation where the data in each class can

be approximated by a Gaussian distribution. When this condition is not satisfied, PPC could lead to poor results. One way to alleviate this weakness is to use multiple clusters to model one class [23]. Third, in choosing the weight matrix w, although our design works well on some data sets, it is not clear how to set the weight for a more general situation.

To address the computational difficulty caused by large cliques, we propose two approximation methods: Gibbs sampling and mean field approximation. We also observe Gibbs sampling can be fairly slow for large cliques. One way to address this problem is to use fewer sampling rounds (and thus a cruder approximate inference) in the early phase of EM training, and gradually increase the number of sampling rounds (and a finer approximation) when EM is close to convergence. By doing this, we may be able to achieve a much faster algorithm without sacrificing too much precision. For the mean field approximation, the bias brought by the independence assumption among $Q_i(\cdot)$ could be severe for some problems. We can ameliorate this, as suggested by [9], by retaining more sub-structure of the original graphical model (for PPC, it is expressed in w), while still keeping the computation tractable.

Appendix A

In this part of appendix, we prove that when $|w(i,j)| \to \infty$ for each specified pair (x_i, x_j), the complete likelihood of PPC can be written as in equation (1.10), and thus equivalent to the model proposed by [16].

In the model proposed by [16], the complete likelihood is written as :

$$\begin{aligned} P(X, Z|\Theta, E_\Omega) &= \frac{1}{\Omega} \prod_{c_i} \delta_{y_{c_i}} \prod_{a_i^1 \neq a_i^2} (1 - \delta_{y_{a_i^1}, y_{a_i^2}}) \prod_{i=1}^{N} P(z_i|\Theta) P(x_i|z_i, \Theta) \\ &= \frac{1}{\Omega} \prod_{c_i} \delta_{y_{c_i}} \prod_{a_i^1 \neq a_i^2} (1 - \delta_{y_{a_i^1} y_{a_i^2}}) P(X, Z|\Theta) \end{aligned}$$

where E_Ω stands for the pairwise constraints, $\delta_{y_{c_i}}$ is 1 iff all the points in the chunklet c_i have the same label, (a_i^1, a_i^2) is the index of the sample pair with hard cannot-link between them. This is equivalent to

$$P(X, Z|\Theta, E_\Omega) = \begin{cases} \frac{1}{\Omega} P(X, Z|\Theta) & Z \text{ satisfies all the constraints;} \\ 0 & \text{otherwise.} \end{cases} \tag{1.29}$$

In the corresponding PPC model with hard constraints, we have

$$w(i,j) = \begin{cases} +\infty & (i,j) \in C_= \\ -\infty & (i,j) \in C_{\neq} \\ 0 & \text{otherwise} \end{cases}$$

According to equation (1.5) and (1.29), to prove $P(X, Z|\Theta, E_\Omega) = P_p(X, Z|\Theta, W)$, we only need to prove $P_p(Z|\Theta, W) = 0$ for all the Z that violate the constraints, that is

$$P_p(Z|\Theta, W) = \frac{\prod_k \pi_{z_k} \prod_{i\neq j} \exp(w(i,j)\ \delta_{z_i z_j})}{\sum_Z \prod_l \pi_{z_l} \prod_{m\neq n} \exp(w(m,n)\ \delta_{z_m z_n)}} = 0.$$

First let us assume Z violates one must-link between pair (α, β) ($w(\alpha, \beta) = +\infty$), we have

$$z_\alpha \neq z_\beta \Rightarrow \delta_{z_\alpha z_\beta} = 0 \Rightarrow \exp(w(\alpha, \beta)\ \delta_{z_\alpha z_\beta)} = 1.$$

We assume the constraints are consistent. In other words, there is at least one Z that satisfies all the constraints. We can denote one such Z by Z^*. We also assume each component has a positive prior probability. It is straightforward to show that

$$P_p(Z^*|\Theta, W) > 0.$$

Then it is easy to show

$$\begin{aligned}
P_p(Z|\Theta, W) &= \frac{\prod_k \pi_{z_k} \prod_{i\neq j} \exp(w(i,j)\ \delta_{z_i z_j})}{\sum_Z \prod_l \pi_{z_l} \prod_{m\neq n} \exp(w(m,n)\ \delta_{z_m, z_n})} \\
&\leq \frac{\prod_k \pi_{z_k} \prod_{i\neq j} \exp(w(i,j)\ \delta_{z_i z_j})}{\prod_k \pi_{z_k^*} \prod_{i\neq j} \exp(w(m,n)\ \delta_{z_i^* z_j^*})} \\
&= (\prod_k \frac{\pi_{z_k}}{\pi_{z_k^*}} \prod_{(i,j)\neq(\alpha,\beta)} \frac{\exp(w(i,j)\ \delta_{z_i z_j})}{\exp(w(i,j)\ \delta_{z_i^* z_j^*})}) \frac{\exp(2w(\alpha,\beta)\ \delta_{z_\alpha z_\beta})}{\exp(2w(\alpha,\beta)\ \delta_{z_\alpha^* z_\beta^*})} \\
&= (\prod_k \frac{\pi_{z_k}}{\pi_{z_k^*}} \prod_{(i,j)\neq(\alpha,\beta)} \frac{\exp(w(i,j)\ \delta_{z_i z_j})}{\exp(w(i,j)\ \delta_{z_i^* z_j^*})}) \frac{1}{\exp(2w(\alpha,\beta)\ \delta_{z_\alpha^* z_\beta^*})}
\end{aligned}$$

Since Z^* satisfies all the constraints, we must have

$$\prod_{(i,j)\neq(\alpha,\beta)} \frac{\exp(w(i,j)\ \delta_{z_i z_j})}{\exp(w(i,j)\ \delta_{z_i^* z_j^*})} \leq 1.$$

So we have

$$P_p(Z|\Theta, W) \leq (\prod_k \frac{\pi_{z_k}}{\pi_{z_k^*}}) \frac{1}{\exp(2w(\alpha,\beta)\ \delta_{z_\alpha^* z_\beta^*})}.$$

When $w(\alpha, \beta) \to +\infty$, we have

$$\frac{1}{\exp(2w(\alpha,\beta)\ \delta_{z_\alpha^* z_\beta^*})} \to 0$$

and then $P_p(Z|\Theta, W) \leq (\prod_k \frac{\pi_{z_k}}{\pi_{z_k^*}}) \frac{1}{\exp(2w(\alpha,\beta)\ \delta_{z_\alpha^* z_\beta^*})} \to 0$. The cannot-link case can be proven in a similar way.

■

Appendix B

In this appendix, we show how to derive weight w from the certainty value γ_{ij} for each pair (x_i, x_j). Let E denote those original (noise-free) labeled pairwise relations and $\tilde{E}$ the noisy version delivered to us. If we know the original pairwise relations E, we only have to consider the cluster assignments that are consistent with E and neglect the others, that is, the prior probability of Z is

$$P(Z|\Theta, E) = \begin{cases} \frac{1}{\Omega_E} P(Z|\Theta) & Z \text{ is consistent with } E \\ 0 & \text{otherwise,} \end{cases}$$

where Ω_E is the normalization constant for E: $\Omega_E = \sum_{Z:\text{ consistent with } E} P(Z|\Theta)$. Since we know $\tilde{E}$ and the associated certainty values $\Gamma = \{\gamma_{ij}\}$, we know

$$P(Z|\Theta, \tilde{E}, \Gamma) = \sum_E P(Z|\Theta, E, \tilde{E}, \Gamma) P(E|\tilde{E}, \Gamma) \tag{1.30}$$

$$= \sum_E P(Z|\Theta, E) P(E|\tilde{E}, \Gamma). \tag{1.31}$$

Let $E(Z) \equiv$ the unique E that is consistent with Z, from equation (1.31) we know

$$\begin{aligned} P(Z|\Theta, \tilde{E}, \Gamma) &= P_p(Z|\Theta, E(Z)) P(E(Z)|\tilde{E}, \Gamma) \\ &= \frac{1}{\Omega_E} P(Z|\Theta) P(E(Z)|\tilde{E}, \Gamma) = \frac{1}{\Omega_E} P(E(Z)|\tilde{E}, \Gamma) P(Z|\Theta). \end{aligned}$$

If we ignore the variation of Ω_E over E, we can get an approximation of $P(Z|\Theta, \tilde{E}, \Gamma)$, denoted as $P_a(Z|\Theta, \tilde{E}, \Gamma)$:

$$\begin{aligned} P_a(Z|\Theta, \tilde{E}, \Gamma) &= \frac{1}{\Omega_a} P(Z|\Theta) P(E(Z)|\tilde{E}, \Gamma) \\ &= \frac{1}{\Omega_a} P(Z|\Theta) \prod_{i<j} \gamma_{ij}^{H_{ij}(\tilde{E}, z_i, z_j)} (1 - \gamma_{ij})^{1 - H_{ij}(\tilde{E}, z_i, z_j)} \end{aligned}$$

where Ω_a is the new normalization constant: $\Omega_a = \sum_Z P(Z|\Theta) P(E(Z)|\tilde{E}, \Gamma)$ and

$$H_{ij}(\tilde{E}, z_i, z_j) = \begin{cases} 1 & (z_i, z_j) \text{ is consistent with } \tilde{E} \\ 0 & \text{otherwise} \end{cases}.$$

We argue that $P_a(Z|\Theta, \tilde{E}, \Gamma)$ is equal to a PPC prior probability $P_p(Z|\Theta, W)$ with

$$w(i,j) = \begin{cases} \frac{1}{2} \log(\frac{\gamma_{ij}}{1-\gamma_{ij}}) & (z_i, z_j) \text{ is specified as must-linked in } \tilde{E} \\ -\frac{1}{2} \log(\frac{\gamma_{ij}}{1-\gamma_{ij}}) & (z_i, z_j) \text{ is specified as cannot-linked in } \tilde{E} \\ 0 & \text{otherwise.} \end{cases} \tag{1.32}$$

This can be easily proven by verifying

$$\frac{P_p(Z|\Theta,W)}{P_a(Z|\Theta,\tilde{E},\Gamma)} = \frac{\Omega_a}{\Omega_w}\prod_{i<j,w(i,j)\neq 0}\gamma_{ij}^{\text{sign}(w(i,j))-1}(1-\gamma_{ij})^{-\text{sign}(w(i,j))} = \text{constant}.$$

Since both $P_a(Z|\Theta,\tilde{E},\Gamma)$ and $P_p(Z|\Theta,W)$ are normalized, we know

$$P_a(Z|\Theta,\tilde{E},\Gamma) = P_p(Z|\Theta,W).$$

Appendix C

In this appendix, we show how to derive K-means model with soft and hard constraints from PPC.

C.1 From PPC to K-means with soft constraints

The adopted cost function for K-means with soft constraints is:

$$J(\mu,Z) = \frac{1}{2}\sum_{i=1}^{N}||x_i-\mu_{z_i}||^2 + \sum_{(i,j)\in C_=}a_{ij}(z_i\neq z_j) + \sum_{(i,j)\in C_\neq}b_{ij}(z_i=z_j) \quad (1.33)$$

where μ_k is the center of the k^{th} cluster. Equation (1.22) can be rewritten as

$$J(\mu,Z) = \frac{1}{2}\sum_{i=1}^{N}||x_i-\mu_{z_i}||^2 - \sum_{ij}w(i,j)\delta_{z_iz_j} + C, \quad (1.34)$$

with $C = -\sum_{(i,j)\in C_=}a_{ij}$ is a constant and

$$w(i,j) = \begin{cases} a_{ij} & (i,j)\in C_= \\ -b_{ij} & (i,j)\in C_\neq \\ 0 & \text{otherwise.}\end{cases} \quad (1.35)$$

The clustering process includes minimizing the cost function $J(\mu,Z)$ over both the model parameters $\mu=\{\mu_1,\mu_2,...,\mu_M\}$ and cluster assignment $Z=\{z_1,z_2,...,z_N\}$. The optimization is usually done iteratively with modified Linde-Buzo-Gray (LBG) algorithm. Assume we have the PPC model with the matrix w same as in equation (1.34). We further constrain each Gaussian component to be spherical with radius σ. The complete data likelihood for PPC model is

$$P(X,Z|\Theta,W) = \frac{1}{\Omega}\prod_{i=1}^{N}\{\pi_{z_i}\exp(-\sum_{i=1}^{N}\frac{||x_i-\mu_{z_i}||^2}{2\sigma^2})\}\prod_{mn}\exp(w(m,n)\delta_{z_mz_n}), \quad (1.36)$$

where Ω is the normalizing constant and μ_k is the mean of the k^{th} Gaussian component. To build its connection to the cost function in equation (1.34), we consider the following scaling:

$$\sigma \to \alpha\sigma, \quad w(i,j) \to w(i,j)/\alpha^2. \tag{1.37}$$

The complete data likelihood with the scaling parameters α is

$$P_\alpha(X, Z|\Theta, W) = \frac{1}{\Omega(\alpha)} \prod_{i=1}^{N} \{\pi_{z_i} \exp(-\sum_{i=1}^{N} \frac{||x_i - \mu_{z_i}||^2}{2\alpha^2\sigma^2})\} \prod_{mn} \exp(\frac{w(m,n)}{\alpha^2}\delta_{z_m z_n}). \tag{1.38}$$

It can be shown that when $\alpha \to 0$, the maximum data likelihood will dominate the data likelihood

$$\lim_{\alpha\to 0} \frac{\max_Z P_\alpha(X, Z|\Theta, W)}{\sum_Z P_\alpha(X, Z|\Theta, W)} = 1. \tag{1.39}$$

To prove equation (1.39), we first show that when α is small enough, we have

$$\arg\max_Z P_\alpha(X, Z|\Theta, W) = Z^* \equiv \arg\min_Z \{\sum_{i=1}^{N} \frac{||x_i - \mu_{z_i^*}||^2}{2} - \sum_{mn} w(m,n)\delta_{z_m^* z_n^*}\}. \tag{1.40}$$

Proof of equation (1.40): Assume Z' is any cluster assignment different than Z^*. We only need to show that when α is small enough,

$$P_\alpha(X, Z^*|\Theta, W) > P_\alpha(X, Z'|\Theta, W). \tag{1.41}$$

To prove equation (1.41), we notice that

$$\begin{aligned} &\log P_\alpha(X, Z^*|\Theta, W) - \log P_\alpha(X, Z'|\Theta, W) \\ &= \sum_{i=1}^{N} (\log \pi_{z_i^*} - \log \pi_{z_i'}) + \frac{1}{\alpha^2}\{\sum_{i=1}^{N} (\frac{||x_i - \mu_{z_i'}||^2}{2} - \frac{||x_i - \mu_{z_i^*}||^2}{2}) - \\ &\qquad \sum_{mn} w(m,n)(\delta_{z'_m z'_n} - \delta_{z_m^* z_n^*})\}. \end{aligned} \tag{1.42}$$

Since $Z^* = \arg\min_Z \{\sum_{i=1}^{N} \frac{||x_i - \mu_{z_i^*}||^2}{2} - \sum_{mn} w(m,n)\delta_{z_m^* z_n^*}\}$, we have

$$\sum_{i=1}^{N} (\frac{||x_i - \mu_{z_i'}||^2}{2} - \frac{||x_i - \mu_{z_i^*}||^2}{2}) - \sum_{mn} w(m,n)(\delta_{z'_m z'_n} - \delta_{z_m^* z_n^*}) > 0. \tag{1.43}$$

Let $\varepsilon = \sum_{i=1}^{N} (\frac{||x_i - \mu_{z_i'}||^2}{2} - \frac{||x_i - \mu_{z_i^*}||^2}{2}) - \sum_{mn} w(m,n)(\delta_{z'_m z'_n} - \delta_{z_m^* z_n^*})$, we can see that when α is small enough

$$\log P_\alpha(X, Z^*|\Theta, W) - \log P_\alpha(X, Z'|\Theta, W) = \sum_{i=1}^{N} (\log \pi_{z_i^*} - \log \pi_{z_i'}) + \frac{\varepsilon}{\alpha^2} > 0. \tag{1.44}$$

■

It is obvious from equation (1.44) that for any Z' different than Z^*

$$\begin{aligned}
&\lim_{\alpha\to 0} \log P_\alpha(X, Z^*|\Theta, W) - \log P_\alpha(X, Z'|\Theta, W) \\
&= \lim_{\alpha\to 0} \sum_{i=1}^{N} (\log \pi_{z_i^*} - \log \pi_{z_i'}) + \frac{\varepsilon}{\alpha^2} \\
&= +\infty,
\end{aligned}$$

or equivalently

$$\lim_{\alpha\to 0} \frac{P_\alpha(X, Z'|\Theta, W)}{P_\alpha(X, Z^*|\Theta, W)} = 0, \tag{1.45}$$

which proves equation (1.39). As the result of equation (1.39), when optimizing the model parameters we can equivalently maximize $\max_Z P_\alpha(X, Z|\Theta, W)$ over Θ. It is then a joint optimization problem

$$\max_{\Theta, Z} P_\alpha(X, Z|\Theta, W).$$

Following the same thought, we find the soft posterior probability of each sample (as in conventional mixture model) becomes hard membership (as in K-means). This fact can be simply proved as follows. The posterior probability of sample x_i to component k is

$$P_\alpha(z_i = k|X, \Theta, W) = \frac{\sum_{Z|z_i=k} P_\alpha(X, Z|\Theta, W)}{\sum_Z P_\alpha(X, Z|\Theta, W)}.$$

From equation (1.39)), it is easy to see

$$\lim_{\alpha\to 0} P_\alpha(z_i = k|X, \Theta, W) = \begin{cases} 1 & z_i^* = k \\ 0 & \text{otherwise.} \end{cases} \tag{1.46}$$

The negative logarithm of the complete likelihood P_α is then:

$$\begin{aligned}
J_\alpha(\Theta, Z) &= -\log P_\alpha(X, Z|\Theta, W) \\
&= -\sum_{i=1}^{N} \log \pi_{z_i} + \sum_{i=1}^{N} \frac{||x_i - \mu_{z_i}||^2}{2\alpha^2} - \sum_{mn} \frac{w(m,n)}{\alpha^2} \delta_{z_m z_n} + \log(\Omega(\alpha)) \\
&= -\sum_{i=1}^{N} \log \pi_{z_i} + \frac{1}{\alpha^2} \left(\sum_{i=1}^{N} \frac{||x_i - \mu_{z_i}||^2}{2} - \sum_{mn} w(m,n) \delta_{z_m z_n}\right) + C,
\end{aligned}$$

where $C = \log \Omega(\alpha)$ is a constant. It is obvious that when $\alpha \to 0$, we can neglect the term $-\sum_{i=1}^{N} \log \pi_{z_i}$. Hence the only model parameters left for

adjusting are the Gaussian means μ. We only have to consider the new cost function

$$\tilde{J}_\alpha(\mu, Z) = \frac{1}{\alpha^2}(\sum_{i=1}^{N} \frac{||x_i - \mu_{z_i}||^2}{2} - \sum_{mn} w(m,n)\delta_{z_m z_n}), \tag{1.47}$$

the optimization of which is obviously equivalent to equation (1.33). So we can conclude that when $\alpha \to 0$ in equation (1.37), the PPC model shown in equation (1.36) becomes a K-means model with soft constraints.

C.2 From PPC to K-means with hard constraints (COPK-means)

COPK-means is a hard clustering algorithm with hard constraints. The goal is to find a set of cluster centers μ and clustering result Z that minimizes the cost function

$$\sum_{i=1}^{N} ||x_i - \mu_{z_i}||^2, \tag{1.48}$$

while subject to the constraints

$$z_i = z_j, \text{ if } (x_i, x_j) \in C_= \tag{1.49}$$
$$z_i \neq z_j, \text{ if } (x_i, x_j) \in C_{\neq}. \tag{1.50}$$

Assume we have the PPC model with soft relations represented with the matrix w such that:

$$w(i,j) = \begin{cases} w & (x_i, x_j) \in C_= \\ -w & (x_i, x_j) \in C_{\neq} \\ 0 & \text{otherwise} \end{cases} \tag{1.51}$$

where $w > 0$. We further constrain each Gaussian component to be spherical with radius σ. The complete data likelihood for PPC model is

$$P(X, Z|\Theta, W) = \frac{1}{\Omega}\prod_{i=1}^{N}\{\pi_{z_i}\exp(-\sum_{i=1}^{N}\frac{||x_i - \mu_{z_i}||^2}{2\sigma^2})\} \prod_{(m,n)\in C_=}\exp(w\delta_{z_m z_n}) \prod_{(m',n')\in C_{\neq}}\exp(-w\delta_{z_{m'} z_{n'}}), \tag{1.52}$$

where μ_k is the mean of the k^{th} Gaussian component. There are infinite ways to get equation (1.48)-(1.50) from equation (1.52), but we consider the following scaling with factor β:

$$\sigma \to \beta\sigma, \;\; w(i,j) \to w(i,j)/\beta^3. \tag{1.53}$$

The complete data likelihood with the scaled parameters is

$$P_\beta(X, Z|\Theta, W) = \frac{1}{\Omega(\beta)} \prod_{i=1}^{N} \{\pi_{z_i} \exp(-\sum_{i=1}^{N} \frac{||x_i - \mu_{z_i}||^2}{2\beta^2\sigma^2})\}$$
$$\prod_{(m,n)\in C_=} \exp(\frac{w}{\beta^3}\delta_{z_m z_n}) \prod_{(m',n')\in C_\neq} \exp(-\frac{w}{\beta^3}\delta_{z_{m'} z_{n'}}), \quad (1.54)$$

As established in C.1, when $\beta \to 0$, the maximum data likelihood will dominate the data likelihood

$$\lim_{\beta\to 0} \frac{\max_Z P_\beta(X, Z|\Theta, W)}{\sum_Z P_\beta(X, Z|\Theta, W)} = 1.$$

As the result, when optimizing the model parameters Θ we can equivalently maximize $\max_Z P_\beta(X, Z|\Theta, W)$. Also, the soft posterior probability (as in conventional mixture model) become hard membership (as in K-means).

The negative logarithm of the complete likelihood P_β is then:

$$J_\beta(\Theta, Z) = -\sum_{i=1}^{N} \log \pi_{z_i} + C +$$
$$\frac{1}{\beta^2}(\sum_{i=1}^{N} \frac{||x_i - \mu_{z_i}||^2}{2} + \frac{1}{\beta}(\sum_{(m',n')\in C_\neq} w\delta_{z_{m'} z_{n'}} - \sum_{(m,n)\in C_=} w\delta_{z_m z_n})), \quad (1.55)$$

where $C = \log \Omega(\beta)$ is a constant. It is obvious that when $\beta \to 0$, we can neglect the term $-\sum_{i=1}^{N} \log \pi_{z_i}$. Hence we only have to consider the new cost function

$$\tilde{J}_\beta(\mu, Z) = \frac{1}{\beta^2}(\sum_{i=1}^{N} \frac{||x_i - \mu_{z_i}||^2}{2} + \frac{1}{\beta}(\sum_{(m',n')\in C_\neq} w\delta_{z_{m'} z_{n'}} - \sum_{(m,n)\in C_=} w\delta_{z_j, z_k})), \quad (1.56)$$

the minimization of which is obviously equivalent to the following equation since we can neglect the constant factor $\frac{1}{\beta^2}$:

$$\tilde{\tilde{J}}_\beta(\mu, Z) = \sum_{i=1}^{N} \frac{||x_i - \mu_{z_i}||^2}{2} + \frac{w}{\beta} J_c(Z). \quad (1.57)$$

where $J_c(Z) = \sum_{(m',n')\in C_\neq} \delta_{z_{m'} z_{n'}} - \sum_{(m,n)\in C_=} \delta_{z_m z_n}$ is the cost function term from pairwise constraints.

Let $S_Z = \{Z | z_i = z_j \text{ if } w(i,j) > 0; z_i \neq z_j \text{ if } w(i,j) < 0;\}$. We assume the pairwise relations are consistent, that is, $S_Z \neq \emptyset$. Obviously, all Z in S_Z achieve the same minimum value of the term $J_c(Z)$. That is

$$\forall Z \in S_Z, Z' \in S_Z \;\; J_c(Z) = J_c(Z')$$
$$\forall Z \in S_Z, Z'' \notin S_Z \;\; J_c(Z) < J_c(Z'').$$

It is obvious that when $\beta \to 0$, any Z that minimizes $\tilde{\tilde{J}}_\beta(\mu, Z)$ must be in S_Z. So the minimization of equation (1.54) can be finally casted into the following form:

$$\begin{aligned} \min_{Z,\mu} \quad & \sum_{i=1}^{N} ||x_i - \mu_{z_i}||^2 \\ \text{subject to} \quad & Z \in S_Z, \end{aligned}$$

which is apparently equivalent to equation (1.48)-(1.50). So we can conclude that $\beta \to 0$ in equation (1.53), the PPC model shown in equation (1.52) becomes a K-means model with hard constraints.

References

[1] C. Ambroise, M. Dang, and G. Govaert. Clustering of spatial data by the EM algorithm. In A. Soares, J. Gmez-Hernndez, and R. Froidevaux, editors, *Geostatistics for Environmental Applications*, volume 3, pages 493–504. Kluwer, 1997.

[2] S. Basu, A. Bannerjee, and R. Mooney. Semi-supervised clustering by seeding. In C. Sammut and A. Hoffmann, editors, *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 19–26. Morgan Kaufmann, 2002.

[3] S. Basu, M. Bilenko, and R. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 59–68, 2004.

[4] M. Bilenko, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In C. Brodley, editor, *Proceedings of the Twenty-first International Conference on Machine Learning*, pages 11–18. ACM, 2004.

[5] C. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IEEE Transaction on Image Processing*, 3:162–177, March 1994.

[6] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised Clustering with User Feedback. Technical Report TR2003-1892, Cornell University, 2003.

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

[8] H. Wang E. Segal and D Koller. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, 19, 2003.

[9] T. Jaakkola. Tutorial on variational approximation methods. In C. Brodley, editor, *Proceedings of the Twenty-first International Conference on Machine Learning*, pages 11–18. ACM, 2004.

[10] D. Klein, S. Kamvar, and C. Manning. From instance level to space-level constraints: making the most of prior knowledge in data clustering.

0-8493-0052-5/00/$0.00+$.50

In C. Sammut and A. Hoffmann, editors, *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 307–313. Morgan Kaufmann, 2002.

[11] T. Lange, M. Law, A. Jain, and J. Buhmann. Learning with constrained and unlabelled data. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 730–737, 2005.

[12] M. Law, A. Topchy, and A. Jain. Clustering with soft and group constraints. In A. Fred, T. Caelli, R. Duin, A. Campilho, and D. Ridder, editors, *Joint IAPR International Workshop on Syntactical and Structural Pattern Recognition and Statistical Pattern Recognition*, pages 662–670. Springer-Verlag, 2004.

[13] M. Law, A. Topchy, and A. Jain. Model-based clustering with probabilistic constraints. In *Proceedings of SIAM Data Mining*, pages 641–645, 2005.

[14] Z. Lu and T. Leen. Semi-supervised clustering with pairwise constraints: A discriminative approach. In *Eleventh International Conference on Artificial Intelligence and Statistics*. 2007.

[15] R. Neal. Probabilistic inference using Markov Chain Monte Carlo methods. Technical Report CRG-TR-93-1, Computer Science Department, Toronto University, 1993.

[16] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using side-information. Technical Report 2003-43, Leibniz Center for Research in Computer Science, 2003.

[17] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing Gaussian mixture models with EM using equivalence constraints. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing System*, volume 16, pages 505–512. 2004.

[18] A. Srivastava and J. Stroeve. Onboard Detection of Snow, Ice and Other Geophysical Processes Using Kernel Methods. In *ICML 2003 Workshop on Machine Learning Technologies for Autonomous Space Sciences*, 2003.

[19] J. Theiler and G. Gisler. A contiguity-enhanced K-means clustering algorithm for unsupervised multispectral image segementation. In B. Javidi and D. Psaltis, editors, *Proceedings of SPIE*, volume 3159, pages 108–118. SPIE, 1997.

[20] K. Wagstaff. *Intellgient clustering with instance-level constraints*. PhD thesis, Cornell University, 2002.

[21] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained K-means clustering with background knowledge. In C. Brodley and

A. Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584. Morgan Kaufmann, 2001.

[22] E. Xing, A. Ng, M. Jordan, and S. Russe. Distance metric learning with applications to clustering with side information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing System*, volume 15, pages 505–512. Cambridge, MA: MIT Press, 2003.

[23] Q. Zhao and D. Miller. Mixture modeling with pairwise, instance-level class constraints. *Neural Computation*, 17, 2005.