Revisiting Probabilistic Models for Clustering with Pair-wise Constraints

Blaine Nelson, Ira Cohen
Enterprise Systems and Software Laboratory
HP Laboratories Palo Alto
HPL-2007-79
May 8, 2007*

We revisit recently proposed algorithms for probabilistic clustering with pair-wise constraints between data points. We evaluate and compare existing techniques in terms of robustness to misspecified constraints. We show that the technique that strictly enforces the given constraints, namely the chunklet model, produces poor results even under a small number of misspecified constraints. We further show that methods that penalize constraint violation are more robust to misspecified constraints but have undesirable local behaviors. Based on this evaluation, we propose a new learning technique, extending the chunklet model to allow soft constraints represented by an intuitive measure of confidence in the constraint.

* Internal Accession Date Only
Presented at the 24th International Conference on Machine Learning, 20-24 June 2007, Corvallis, OR, USA
Approved for External Publication

Revisiting Probabilistic Models for Clustering with Pair-wise Constraints

Blaine Nelson NELSONB@EECS.BERKELEY.EDU
University of California, Berkeley, Department of EECS, Berkeley, CA

Ira Cohen IRA.COHEN@HP.COM
HP Research Labs, Palo Alto, CA

Abstract

We revisit recently proposed algorithms for probabilistic clustering with pair-wise constraints between data points. We evaluate and compare existing techniques in terms of robustness to misspecified constraints. We show that the technique that strictly enforces the given constraints, namely the chunklet model, produces poor results even under a small number of misspecified constraints. We further show that methods that penalize constraint violation are more robust to misspecified constraints but have undesirable local behaviors. Based on this evaluation, we propose a new learning technique, extending the chunklet model to allow soft constraints represented by an intuitive measure of confidence in the constraint.

1. Introduction

Clustering is the traditional problem of learning a partition of an observed dataset $X = \{x_i\}_{i=1}^N$ of N data points into K clusters. The traditional goal is to choose a partitioning $Y = \{y_i \in \{1 \dots K\}\}_{i=1}^N$ that optimizes an objective function $\mathfrak{J}(X, Y)$; e.g., minimizing intra-cluster variance. However, such broad clustering objectives are not necessarily congruent with the particular notion of separation for any given task. This has motivated the incorporation of prior knowledge to guide the clustering process toward a desirable partition. One form of prior knowledge is pair-wise constraints among a subset of data points.

In recent years, clustering with pair-wise constraints emerged as a new paradigm for semi-supervised clustering. In this framework, the clustering agent is given observations X and a set of constraints $\mathbb{C}$ composed of pair-wise **must-link** and **cannot-link** constraints specifying points that should or should not be clustered together, respectively. These constraints are typically assumed to be either given by an expert or inferred from domain knowledge. There are two primary strategies for incorporating must- and cannot-link constraints: learning a metric and constrained clustering. This work only considers constrained clustering.

Constrained clustering techniques directly incorporate constraints into the clustering procedure. Some constrained clustering algorithms use modifications to graph-based techniques (Yu & Shi, 2001; Kamvar et al., 2003; Kulis et al., 2005). Other techniques explicitly used the constraints to reduce the search space of common clustering algorithms (Wagstaff & Cardie, 2000; Wagstaff et al., 2001). More recent techniques incorporate the constraints directly into their models, resulting in probabilistic models that augment mixture models by directly modeling the constraints (Shental et al., 2003; Basu et al., 2004; Lu & Leen, 2004; Lange et al., 2005).

In this paper we revisit probabilistic mixture models for clustering with pairwise constraints, highlighting the positive and negative aspects of existing techniques. In particular, we focus on three main issues. The first is the robustness of the various methods to the realistic case in which some constraints are misspecified. The second is the difficulty in specifying interpretable penalty weights. The third is the local nature of approximate inference methods, which can lead to suboptimal results. Further, we introduce a new approximation algorithm that extends the chunklet model (Shental et al., 2003) to soft constraints, thereby providing robustness against misspecified constraints.

The rest of the paper is organized as follows. In Section 2 we describe the probabilistic models for clustering with pair-wise constraints and we critique them in Section 3. In Section 4 we present our approach—an extension of the chunklet model. We empirically compare these approaches in Section 5 followed by a discussion in Section 6.

Appearing in *Proceedings of the 24^{th} International Conference on Machine Learning*, Corvallis, OR, 2007.

2. Background

We consider the problem of clustering a set X of data points into K clusters using a model parameterized by Θ (e.g. a set of centroids $\{\mu_c\}$) with side information in the form of a set of constraints $\mathbb{C}$ composed of pairwise **must-link** and **cannot-link** constraints. Each must-link constraint $(i \leftrightarrow j)$ specifies that the points i and j should be in the same cluster. Similarly, every cannot link $(k \nleftrightarrow \ell)$ suggests k and ℓ should not be in the same cluster. Constraints that are strictly enforced are called **absolute**; that is, any configuration that violates an absolute constraint has zero probability. Constraints that are can be violated are called **soft** constraints, and they also have an associated violation penalty. Here $W_{ij} \in [0, +\infty]$ denotes the penalty for violating a soft constraint between i and j.[1] In the remainder of this paper, constraints will be soft unless otherwise specified.

In this work, we only consider probabilistic models that extend the mixture model framework; e.g., a **mixture of Gaussians**. In a classical mixture model, there are two independence assumptions: (1) given the model's parameters Θ, all labels are independent, and (2) given its label y_i, the data point x_i is independent of all other labels and data. Formally, these are

$$\mathrm{P}\left(Y|\Theta\right) = \prod_{i=1}^{N} \mathrm{P}\left(y_i|\Theta\right) \quad (1)$$

$$\mathrm{P}\left(X|Y,\Theta\right) = \prod_{i=1}^{N} \mathrm{P}\left(x_i|y_i,\Theta\right). \quad (2)$$

These assumptions define the fundamental components of the mixture model: the **prior distribution** on the labels and the **data model**.

We consider probabilistic models that extend classic mixture models by constructing a **hidden Markov random field** (HMRF) on the labels (Basu et al., 2004). In an HMRF, the must-link and cannot-link constraints are represented graphically by undirected links between labels and the graph is assumed to be **Markovian**: the distribution of a label only depends only on its **neighborhood** $\mathcal{N}_i \triangleq \{j \mid (i,j) \in \mathbb{C}\}$. Thus, for the HMRF the prior distribution satisfies

$$\mathrm{P}\left(y_i|Y_{-i},\Theta,\mathbb{C}\right) = \mathrm{P}\left(y_i|Y_{\mathcal{N}_i},\Theta,\mathbb{C}\right), \quad (3)$$

where Y_{-i} denotes the set of all labels other than y_i.

An HMRF violates the independence assumption in Eq. (1) but preserves the data model in Eq. (2). The new prior distribution that replaces Eq. (1) is

$$\mathrm{P}\left(Y|\Omega_{\mathbb{C}},\Theta,\mathbb{C}\right) \propto \mathrm{P}\left(Y|\Theta\right)\mathrm{P}\left(\Omega_{\mathbb{C}}|Y,\Theta,\mathbb{C}\right), \quad (4)$$

where $\Omega_{\mathbb{C}}$ is the event that Y is consistent with the constraints. Here, $\mathrm{P}\left(Y|\Theta\right)$ is the original prior given in Eq. (1) and $\mathrm{P}\left(\Omega_{\mathbb{C}}|Y,\Theta,\mathbb{C}\right)$ is a **weighting function** for constraint violations. The form of this weighting function is a direct consequence of the HMRF structure. The Hammersley-Clifford theorem shows that the HMRF's Markovian assumption is equivalent to a Gibbs distribution. The particular form chosen is defined by a penalty V_{ij} as

$$\mathrm{P}\left(\Omega_{\mathbb{C}}|Y,\Theta,\mathbb{C}\right) \propto \exp\left\{-\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i} V_{ij}\right\} \quad (5)$$

$$V_{ij} = \begin{cases} -\mathbb{I}\left\{y_i = y_j\right\}\cdot W_{ij} & i \leftrightarrow j \\ \mathbb{I}\left\{y_i = y_j\right\}\cdot W_{ij} & i \nleftrightarrow j \\ 0 & \text{o.w.} \end{cases} \quad (6)$$

where $\mathbb{I}\left\{\cdot\right\}$ is the indicator function of an event.[2] Finally, the conditional distribution for the label y_i is

$$\mathrm{P}\left(y_i|Y_{\mathcal{N}_i},\Theta,\mathbb{C}\right) \propto \exp\left\{-2\sum_{j\in\mathcal{N}_i} V_{ij}\right\}. \quad (7)$$

In the remainder of this section, we review techniques that use the Gibbs distribution. All of the techniques considered here are EM-style algorithms. With a few slight variations, the M-steps of all these algorithms are nearly identical—they all estimate cluster parameters Θ using maximum-likelihood estimation procedures. However, the E-steps of these approaches differ depending on the approximate inference methods used to compute the posterior:

$$\mathrm{P}\left(y_i|X,\Theta,\mathbb{C}\right) = \sum_{Y_{-i}} \underbrace{\mathrm{P}\left(X|Y,\Theta\right)}_{\text{data model}} \underbrace{\mathrm{P}\left(Y|\Theta,\mathbb{C}\right)}_{\text{label prior}}. \quad (8)$$

In theory, this posterior can be computed exactly using the junction tree algorithm. This computation is NP-complete in general, but it can be approximated. In this setting, the most prominent approximation approaches are based on the chunklet model (Shental et al., 2003), iterated conditional modes (Basu et al., 2004), Gibbs sampling (Lu & Leen, 2004), and the mean-field approximation (Lange et al., 2005). Next we describe in detail each algorithm and how it differs from the others.

2.1. The Chunklet Model

If we assume that all constraints are correct, we can restrict the problem to sets of only absolute constraints. In this

[1] The case of absolute constraints is equivalent to restricting $W_{ij} \in \{0, +\infty\}$.

[2] Equivalently, we can define the penalty $V_{ij} = (1 - \mathbb{I}\left\{y_i = y_j\right\})\cdot W_{ij}$ when $i \leftrightarrow j$ to make it everywhere positive. However, this alternative only differs by an additive constant in the exponent of Eq. (5) and thus is part of the constant of proportionality.

setting, we can exploit the relational semantics of the constraints to simplify the posterior inference. The resulting method is called the **chunklet model** (Shental et al., 2003).

The case of absolute constraints leads to a Gibbs prior that assigns equal probability to every consistent assignment and zero probability otherwise. Thus, Eq. (5) reduces to

$$\mathrm{P}\left(\Omega_{\hat{\mathbb{C}}}|Y,\Theta,\hat{\mathbb{C}}\right) \propto \mathbb{I}\left\{Y \in \Gamma_{\hat{\mathbb{C}}}\right\}$$

where $\hat{\mathbb{C}}$ is the set of absolute constraints and $\Gamma_{\hat{\mathbb{C}}}$ is the set of assignments consistent with them. Moreover, since all constraints are satisfied, the chunklet model exploits the properties of the constraints. In particular, the must-link is an equivalence relation and induces a partition of the data points into equivalence classes called **chunklets**—a set of points that belong to the same cluster. The chunklets are defined by the transitive closure of the must-link graph and the L-th chunklet is modeled as a group of points with a single common label, y_L^{ch}. Thus, the analog of Eq. (8) for a chunklet is

$$\mathrm{P}\left(y_L^{ch}|X,\Theta\right) = \mathrm{P}\left(y_L^{ch}|\Theta\right)\prod_{i\in L}\mathrm{P}\left(x_i|y_L^{ch},\Theta\right) \ .$$

The grouping behavior of the chunklets is desirable in the following sense: the points of a chunklet act as a single entity that penalizes large deviations. For instance, in a K-means setting with a Gaussian data model, maximizing a chunklet's posterior is equivalent to minimizing the sum of the *squared* distance of each point in the chunklet to the clusters' centroids.

Finally, cannot-links also transfer to the chunklets; that is, $i \in L$ and $j \in M$ and $i \nleftrightarrow j \in \hat{\mathbb{C}}$ implies $L \nleftrightarrow M$. Unfortunately, exact inference on chunklets with cannot-links requires the junction-tree algorithm. However, inference on the chunklet's HMRF is often simpler in practice since chunklets partially group the data.

2.2. The ICM Approach

The iterated conditional modes (ICM) approach to the HMRF (Basu et al., 2004) is designed to find an assignment to Y that maximizes the joint probability of the labels given the data, Θ and $\mathbb{C}$, for the purposes of a K-means algorithm with hard assignments; i.e.,

$$\max_{Y}\left[\mathrm{P}\left(Y|X,\Theta,\mathbb{C}\right)\right].$$

To avoid expensive junction tree calculations, ICM performs a local search over the space of possible labels. It begins with the assignment that maximizes the data model $\mathrm{P}\left(X|Y,\Theta\right)$ and iteratively increases the complete joint distribution by greedily changing each label to minimize the sum of its distance to the cluster centroid and its constraint violation penalties. This process continues until unilaterally changing a single label can no longer increase the joint probability. While orders of magnitude faster than most other approaches, the ICM approximation is extremely dependent on the order of label updates and is only appropriate in a K-means setting with hard assignments to clusters.

2.3. The PPC Model

The Penalized Probabilistic Clustering (PPC) algorithm (Lu & Leen, 2004) extends the ICM technique for soft assignments via Gibbs sampling. As with ICM, labels are changed based on the objective function, but PPC chooses assignments probabilistically. In particular, for each sample, its label y_i is sampled conditioned on the current values Y_{-i} for all other labels. Averaging over a set of these samples gives a distribution over the possible assignments for each y_i. In this way, the method approximates the soft assignment probabilities of each point. However, sampling over assignments can be slow.

2.4. The Maximum-Entropy model

Finally, the Maximum-Entropy model performs approximate inference on the weighted HMRF using a mean field approximation (Lange et al., 2005). In particular, this approach constructs a factorial approximate distribution $q(Y) = \prod_i q_i(y_i)$ minimizing its KL divergence to the actual posterior in Eq. (8). This results in a search for a stationary point of $q_i(k)$, constrained such that $\sum_k q_i(k) = 1$, using the following equation:[3]

$$q_i(y_i) \propto \mathrm{P}\left(x_i|y_i,\Theta\right)\exp\left\{\sum_{j\in\mathcal{N}_i}\left(q_j(y_i)-1\right)V_{ij}\right\}. \quad (9)$$

This provides a fast approximation for soft assignments, but computing the stationary points of the above system of equations can be prone to local minima.

3. Bad Modeling Behaviors

We have now seen several elegant approaches to approximating an HRMF. In this section, we discuss some of their shortcomings and motivate a new method.

3.1. Misspecified Constraints

One of the most important practical properties in constrained clustering is robustness to misspecification of constraints by an expert. As with any data, constraints are subject to some degree of inaccuracy depending on the task. However, the techniques used to make algorithms efficient

[3]The authors originally allowed for a wider range of models by replacing $\mathrm{P}\left(x_i|y_i=k,\Theta\right)$ with a more general expression.

or accurate can lead to poor behavior under even small amounts of error in the constraints.

Constraint propagation is particularly sensitive to misspecified constraints. Consider the transitive closure of must-links—a single incorrect constraint between points i and j is propagated to their entire transitive neighborhoods. Thus, the chunklet model, which assumes error-free constraints, is substantially degraded by even small levels of error. The ICM approach can use the transitive closure to generate new constraints unless contradictory constraints are detected. While this error-detection heuristic mitigates the effect of errors on performance, they still have a substantial impact.

The general technique for handling misspecified constraints is to allow constraint violations but penalize them. While the general technique is sound, there is no semantic meaning associated with the weights specified in Eq. 6. These penalty weights are unitless, and their salience is data-dependent. While the weight 50 is larger than 5, it is unclear how much more of an impact the former will have than the latter. Overall, the notion of specifying penalties seems unintuitive and we propose a different notion of constraint weights.

3.2. Downside of local approximations

Aside from its sensitivity to misspecified constraints, the chunklet model utilizes must-link constraints elegantly. Other approximations to the HMRF are less ideal in the following sense: they only perform local updates based on each data point's immediate neighborhood. The ICM, PPC, and mean field approaches all incrementally update each y_i distribution independently until the field converges. These point-wise updates can be trapped in local optima, especially for large weights. The following examples illustrate these points.

Example 1: Consider a 2-cluster problem with two points i and j and a cannot-link between them with weight W: $i \nleftrightarrow j$. Suppose that the unconstrained distributions of i and j are $[0.1, 0.9]$ and $[0.01, 0.99]$, respectively. While both points are initially assigned to cluster 2, strict enforcement of the cannot-link would place them in separate clusters: Which point should be moved to cluster 1? Running exact inference with the junction tree algorithm gives the following:

	W					
	0.01	0.1	1	3	5	∞
$\mathrm{P}(y_i = 2)$	.898	.881	.567	.102	.084	.083
$\mathrm{P}(y_j = 2)$	.989	.988	.960	.918	.917	.917

For small weights, both are likely to be in cluster 2, reflecting their initial probabilities and low penalty for violation. However, as the weight W increases we see the most likely assignment has point j is in cluster 2 and i is in cluster 1. Unfortunately, these distributions are not necessarily reflected by the HMRF approximations—all depend on the update order to some degree. In the ICM approach, both points would start in cluster 2. If point j is updated first, a large W will move j into cluster 1—an equilibrium state for the ICM algorithm but the wrong one. Gibbs sampling, on the other hand, will eventually approach the true distribution. However, as W increases, the sampling distribution becomes increasingly peaked, thereby decreasing the mixing rate of the chain. Similarly, for the mean-field approach, equilibria of Eq. (9) represent progressively poorer approximations to the HMRF as W gets large.

Example 2: All three of these approaches approximate the distribution of highly connected components poorly when their links have large weights. For instance, consider a 2 cluster problem with a clique of L completely connected points with weight W on all its must-links. The first $L-1$ points are nearly evenly split between the clusters with an unconstrained distribution $[.51, .49]$. However, the L-th point is initially distributed as $[0.01, .99]$. Using junction tree, we find that the most-likely assignment places all points in cluster 2. In the iterative approximation methods, the first $L-1$ points start in cluster 1 and the L-th point is in cluster 2. For $L \geq 3$, the ICM approach will never move any of the first $L-1$ points to cluster 2. Further, the L-th point will move to cluster 1 if $W \geq \frac{4.6}{L-1}$. Gibbs sampling allows for non-optimal assignments, but the clique's dense structure highly penalizes moving any of the first $L-1$ points to cluster 2. Thus, for $L > 3$ and W large, it is extremely unlikely to ever reach the state when all points are in cluster 2. These behaviors stand in stark contrast to the result from exact inference.

4. The Sampled Chunklet Algorithm

To address the issues discussed in the previous section, we construct a model that extends the chunklet model to handle soft constraints by directly sampling constraints to build probabilistic chunklets rather than using approximations to the weighted HMRF. This technique is similar to the Swendsen-Wang method (Swendsen & Wang, 1987) used in statistical mechanics to avoid the pitfalls of local updates and recently it has been used for segmenting images (Barbu & Zhu, 2003). These methods were constructed to augment the performance of MCMC based partition algorithms, whereas our setting has both data metrics and an additional set of constraints.

Our approach performs comparably to the chunklet model in an error-free setting, and it provides robustness against misspecified constraints. Moreover, our approach uses a weight representing the expert's confidence in each constraint instead of arbitrary penalty weights.

The essential idea of our approach is to construct a single sample by sampling each constraint based on the expert's confidences and to infer chunklets from the sampled constraints; i.e., a sample from the space of viable chunklets. For each such sample, we apply the regular chunklet method, and we average over all samples to compute the posterior distribution of the labels. In this section, we explain how our sampling strategy is a valid approximation to the posterior, then we explain how we constructed samples in practice.

4.1. Theoretical Basis for Sampling

We validate our sampling technique by showing that the Gibbs distribution given in Eq. (5) can be approximated by sampling and the posterior inference of y_i can be approximated by averaging distributions inferred from sampled constraints. To this end, let us only consider a set of weighted must-links. For $i \leftrightarrow j$ with weight W_{ij}, the penalty function is equivalent to $V_{ij} = (1 - \mathbb{I}\{y_i = y_j\}) \cdot W_{ij}$, as noted in Footnote 2. The weight function then is

$$\mathrm{P}\left(\Omega_{\mathbb{C}}|Y, \Theta, \mathbb{C}\right) \propto \prod_{i,j=1}^{N} \exp\left\{-W_{ij}\right\}^{(1-\mathbb{I}\{y_i=y_j\})}$$

where $W_{ij} = 0$ for $j \notin \mathcal{N}_i$. Let $P_{ij} \triangleq 1 - \exp\{-W_{ij}\}$ and let $(M_{ij}|Y) \sim \mathrm{Ber}\left(P_{ij}\right)$ be i.i.d. Bernoulli random variables representing the random constraint $i \leftrightarrow j$. By definition, $\mathbb{E}\left[1 - M_{ij}|\, Y\right] = \exp\{-W_{ij}\}$ so we can replace the exponential terms with these conditional expectations in the expression above. Further, the y_i and y_j are not random in these expectations since we condition on Y. Moreover, we have $(1 - \mathbb{I}\{y_i = y_j\}) \in \{0, 1\}$ and $\mathbb{E}[X]^i = \mathbb{E}[X^i]$ for any constant $i \in \{0, 1\}$, so we can move the exponent inside the expectation. Finally, it can be shown that these random variables are independent conditioned on Y, so we are able to exchange the product and expectation. We find that the random variable is the indicator of constraint consistency:

$$\begin{aligned}\mathrm{P}\left(\Omega_{\mathbb{C}}|Y, \Theta, \mathbb{C}\right) &\propto \mathbb{E}\left[\prod_{i,j=1}^{N} (1 - M_{ij})^{(1-\mathbb{I}\{y_i=y_j\})} \Big| Y\right] \\ &= \mathbb{E}\left[\mathbb{I}\left\{Y \in \Gamma_{\mathbb{C}}\right\} \Big| Y\right] \quad .\end{aligned}$$

Now, recall from Section 2.1 that this indicator is exactly the weighting function for the chunklet model. Thus, we approximate our weighting function by averaging the chunklet weighting functions for $\mathbb{S}$ sampled sets of constraints:

$$\tilde{P}(\Omega_{\mathbb{C}}|Y, \Theta, \mathbb{C}) = \frac{1}{\mathbb{S}} \sum_{s=1}^{\mathbb{S}} \mathrm{P}\left(\Omega_{\hat{\mathbb{C}}^{(s)}}|Y, \Theta, \hat{\mathbb{C}}^{(s)}\right) \qquad (10)$$

where the presence of a must-link $i \leftrightarrow j$ in the s-th sample is indicated by $M_{ij}^{(s)} \sim \mathrm{Ber}\left(P_{ij}\right)$. Thus, P_{ij} represents the probability that the must-link $i \leftrightarrow j$ occurs in the s-th model.[4]

This approximation $\tilde{P}$ to the Gibbs prior distribution on the labels can also be used in the E-step for an approximate posterior for each label y_i:

$$\tilde{P}\left(y_i|X, \Theta, \mathbb{C}\right) \propto \frac{1}{\mathbb{S}} \sum_{s=1}^{\mathbb{S}} P^{(s)}(y_i|X, \Theta, \hat{\mathbb{C}}^{(s)}) \qquad (11)$$

where each $P^{(s)}$ is calculated using the chunklet model defined by the sampled absolute constraints $\hat{\mathbb{C}}^{(s)}$ generated in the s-th sample. A similar derivation validates the sampling for cannot-links.

4.2. Constraint Sampling

We describe our methodology for building a sample from the weighted constraints, $\mathbb{C}$. For the s-th sample, we first construct a set of must-link constraints independently: $M_{ij}^{(s)} \sim \mathrm{Ber}(P_{ij})$. Each of these $M_{ij}^{(s)}$ indicates the existence of the must-link $i \leftrightarrow j$ in the s-th sample.

As a practical issue, the sampled constraint graph potentially contains contradictions. When contradictions occur in a sample, it becomes infeasible and must be discarded. However, to avoid wasted samples, we detect the set of potential contradictory cannot-links: $\mathfrak{C} = \{(i, j) \mid i \nleftrightarrow j \text{ is contradictory}\}$. Since the sample is only feasible if *none* of the contradictory constraints are sampled, the probability that the sample will be feasible is $\omega_s = \prod_{(i,j)\in\mathfrak{C}} (1 - P_{ij})$. Thus, in Eq. (11) we can *weight* the sample by ω_s to emulate the effect of the contradictions over many samples without discarding the sample. Finally, the remaining cannot-link constraints are independently sampled.

Combining the sampled must-links and cannot-links gives us a set of *hard* constraints for the s-th sample: $\hat{\mathbb{C}}^{(s)} = \{i \leftrightarrow j \mid M_{ij} = 1\} \cup \{i \nleftrightarrow j \mid (i, j) \notin \mathfrak{C} \wedge C_{ij} = 1\}$. The chunklet model is applied using the sampled constraints to obtain the posteriors $P^{(s)}$ and these samples are combined by using a weighted analog of Eq. (11).

5. Experiments

Here we present empirical results from the HMRF approximations on both toy and real data. While many of these algorithms have different capabilities (e.g. cluster prior and covariance estimation), we compared their approximate inference strategies on a level playing field. To this end, each algorithm was constrained to the simple task of centroid estimation and were given the same initial starting point.

[4] In general, this is *not* equivalent to the probability that $y_i = y_j$.

The performance of different algorithms was assessed by calculating the **normalized mutual information** (NMI); a symmetric measure of the dependency between the clustering and the true labels (Strehl et al., 2000).

5.1. Constraint Generation

We randomly generated weighted constraints for our trials. Given a desired number of must-links M, we generate each must-link by randomly selecting a cluster and uniformly selecting two unique points in that cluster to be must-linked. Similarly, for a cannot-link, two unique clusters are selected at random and a point is randomly selected from each. To introduce errors, a fraction E of the constraints are mislabeled; e.g. a must-link would be mislabeled as a cannot-link. Finally, each constraint is annotated with a probability P from a beta distribution. If the constraint is correct, $P \sim \text{Beta}(\alpha, 1)$ for some $\alpha > 1$; otherwise, $P \sim \text{Beta}(1, \beta)$ for some $\beta > 1$. In our experiments we used $\alpha = \beta = 5$. We generated P this way to reflect the underlying assumption: an expert should have lower certainty in erroneous constraints than in correct ones. This assumption is also implicit in the concept of penalizing constraint violations.

While our sampled chunklet model was designed for constraints annotated with certainties (P_{ij}), other approaches are not directly compatible with this prior information. For approaches designed for penalty weights, we showed in Section 4 that sampling a constraint with probability P_{ij} is equivalent to a penalty weight $W_{ij} = -\log(1 - P_{ij})$ in the Gibbs prior— we use this mapping and when these penalty weights are used with a method, we will subscript the method with "$\log P$". As our experiments show, this mapping proves to be a useful representation of the weights.

5.2. Experiment 1: Toy data

This experiment was conducted on a toy dataset consisting of 200 points sampled from each of three bivariate Gaussians. This dataset is depicted in Figure 1. For each of 100 trials, we constructed four random constraint sets corresponding to $E \in \{0, 0.05, 0.10, 0.25\}$ to assess the algorithms at varying levels of error. For each set of random constraints, we began with no constraints and incrementally added them, producing a trace of each algorithm's performance as it receives progressively more information. Furthermore, to reduce the uncertainty inherent in initialization, each algorithm was seeded with the *correct* labels. However, since no mixture of unit-variance Gaussians can fit our data exactly, the models had to converge to suboptimal assignments. Thus, in this experiment we measured how effectively each approximation technique was able to utilize the constraints in choosing a local minimum in the neighborhood of the true partition.

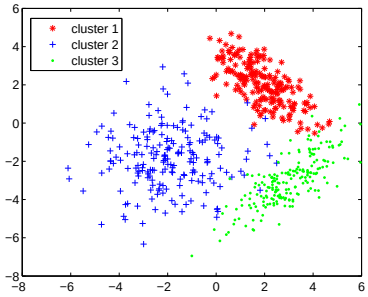
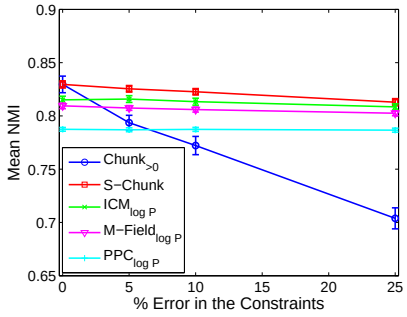


Figure 1. A depiction of the toy data used for the first test. The data consists of 3 distinct but overlapping clusters.

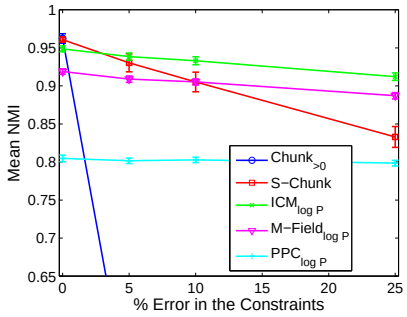
Figure 2 shows the algorithms' performance as a function of the percentage of misspecified constraints for two cases: 100 constraints and 800 constraints. In both cases, with no misspecified constraints, the chunklet and sampled chunklet methods outperform all other methods. However, as the percentage of misspecified constraints increases, the performance of the chunklet model degrades very quickly, while the other methods degrade more gracefully. In addition, with more constraints, all methods perform better when there are no misspecified constraints, but the chunklet method's performance degrades even faster with the increase in specification errors. The sampled chunklet model is more sensitive to the errors compared to the ICM and mean field methods, but performs better with lower error percentages. It is worth noting that while the PPC does not appear to be sensitive to the misspecified constraints, it also hardly utilizes correct constraint when there are no errors.

Finally, Figure 3 shows the performance of the ICM algorithm as a function of the weight set on the constraints. Here we follow the practice used in the original ICM paper, and set a single weight on all constraints. We also extend this practice with a simple heuristic of thresholding the probabilities with a value T; i.e. only using constraints such that $P_{ij} > T$, effectively using fewer constraints when the oracle's confidence is below the threshold, T. Figure 3(a) shows the performance when there are no constraint errors. As the weights increase, there is first an improvement in the ICM's performance, but as the weights further increase, there is significant degradation in performance. Figure 3(b) shows the same results with 25% misspecified constraints. Again, we see that the performance varies depending on the weight used, peaking at weight=1, and dropping off as the weight increases.

The effect of the simple thresholding heuristic is also evident: while the aggressive thresholding resulted in poorer performance when there were no errors ($\text{ICM}_{>0.75}$'s performance is much lower than $\text{ICM}_{>0}$), the opposite is true when there are 25% errors—with the performance of

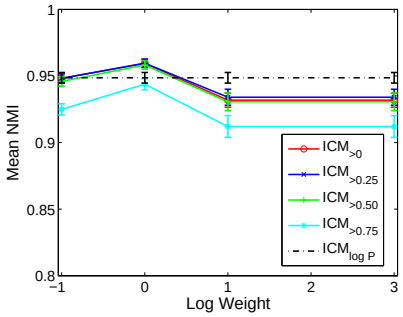


(a) 50 Must-links, 50 Cannot-links

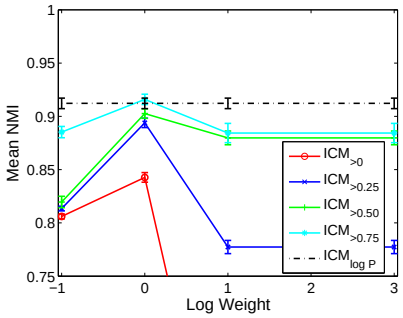


(b) 400 Must-links, 400 Cannot-links

Figure 2. The effect of constraint errors on the various clustering algorithms with 100 (top) and 800 (bottom) constraints. Error bars represent the 99% confidence intervals around the mean NMI.



(a) 0% Error



(b) 25% Error

Figure 3. Mean NMI of ICM approaches with a single weight for all constraints as the weight is increased for different levels of error. Error bars represent the 99% confidence interval. In each trial there were 400 must-link and 400 cannot-link constraints. Baseline using the translated ICM approach ($\text{ICM}_{\log P}$) is used to compare the performance of the uniformly weighted constraints.

$\text{ICM}_{>0}$ and $\text{ICM}_{>0.25}$ degrading significantly as the weight increases. As there is no obvious method for detecting the percentage of misspecified constraints, choosing the threshold T would be difficult. In both figures, we also see that the performance of the ICM with fixed weights compared to the $\text{ICM}_{\log P}$ is lower for most weight settings and thresholds.

5.3. Experiment 2: UCI Datasets

To further illustrate our results, we applied the clustering techniques to two UCI datasets, Digits and Iris (Newman et al., 1998). The Digits datasets is a collection of 2000 instances of the ten digits (200 examples of each digit). The Iris dataset consists of 150 instances of three classes. For the Digits dataset we randomly selected a subset of 400 data points to cluster into ten clusters. We used all the samples of the Iris dataset. In this experiment, we built a 100 trials each with 20 random initial labels and four different levels of error.

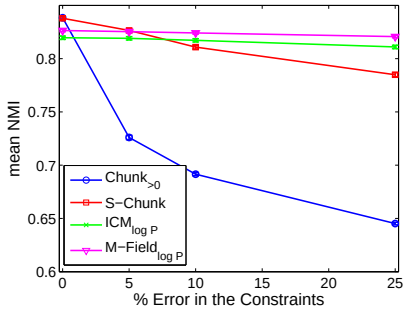
Figure 4 shows the performance of the various algorithms on the two datasets as the percentage of misspecified constraints increases. As observed on the toy data, the chunklet and sampled chunklet methods outperform ICM and mean field when there are no misspecified constraints. However, as the error increases, the chunklet model is the most sensitive to this change, followed by the sampled chunklet model, although it is competitive.

6. Summary and Future Work

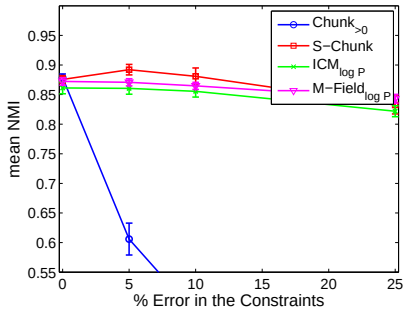
In this paper we revisited probabilistic methods for clustering with pair-wise constraints, highlighting their positive and negative aspects.

We observed that the chunklet model, a direct extension of the mixture of distributions to the case of absolute constraints, is best in an error free setting. It is also highly sensitive to misspecified constraints. In contrast, have also shown that the ICM, PPC and mean field methods, which allow weight-penalized constraint violations, are significantly more robust to misspecified constraints, but are outperformed for low levels of errors.

Additionally, we have shown that methods of approximate inference can produce suboptimal results, depending on the weight on the constraint. Our empirical evidence showed



(a) 150 Must-links, 150 Cannot-links



(b) 100 Must-links, 100 Cannot-links

Figure 4. The effect of misspecified constraints on the Chunklet, sampled chunklets and ICM algorithms for the Digits (top) and Iris (bottom) data.

that this is especially important when the weights are fixed for all constraints. Further, since the scale of penalty weights lacks an intuitive interpretation, choosing their appropriate value poses a practical challenge.

To address these issues we introduced the sampled chunklet algorithm, which extends the chunklet model by sampling constraints. We have shown the theoretical justification for our method and demonstrated that it performs as well as the chunklet model when there are no misspecified constraints. It is also more tolerant of misspecified constraints than the chunklet model though other HMRF approximations tend to outperform our technique for high errors.

In the derivation of our method, we also showed the relationship between the weights used by the existing methods and a probability that represents the expert's confidence in the specified constraints. We further used this relationship to set the weights for the ICM, PPC and mean-field method, with empirical evidence showing that it outperforms both the chunklet model and our sampled chunklet algorithm for high percentages of misspecified constraints.

Finally, by sampling constraints, our method improves the robustness of the chunklet model, but is outperformed by versions of the ICM and mean-field methods in settings with many erroneous constraints. Further investigation suggests that, when in abundance, misspecified constraints contaminate every sample and their effect is compounded by the transitive closure—this leads to poor samples that degrade our technique's performance. Future studies are needed to make our method more robust in these settings.

Acknowledgments

We thank Hewlett Packard for their support of this work. We also thank the ICML reviewers for their comments and suggestions. Finally, we thank Michael Jordan, Anthony Joseph, Peter Bartlett, Marco Barreno, and Junming Yin for their insightful discussions relating to this work.

References

Barbu, A., & Zhu, S.-C. (2003). Graph partition by swendsen-wang cuts. *ICCV*.

Basu, S., Bilenko, M., & Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. *KDD*.

Kamvar, S. D., Klein, D., & Manning, C. D. (2003). Spectral learning. *IJCAI*.

Kulis, B., Basu, S., Dhillon, I. S., & Mooney, R. J. (2005). Semi-supervised graph clustering: a kernel approach. *ICML*.

Lange, T., Law, M. H. C., Jain, A. K., & Buhmann, J. M. (2005). Learning with constrained and unlabelled data. *CVPR*.

Lu, Z., & Leen, T. K. (2004). Semi-supervised learning with penalized probabilistic clustering. *NIPS*.

Newman, D., Hettich, S., Blake, C., & Merz, C. (1998). UCI repository of machine learning databases.

Shental, N., Bar-Hillel, A., Hertz, T., & Weinshall, D. (2003). Computing gaussian mixture models with em using equivalence constraints. *NIPS*. MIT Press.

Strehl, A., Ghosh, J., & Mooney, R. J. (2000). Impact of similarity measures on web-page clustering. *AAAI*.

Swendsen, R. H., & Wang, J.-S. (1987). Nonuniversal critical dynamics in monte carlo simulations. *Phys. Rev. Lett.*, *58*, 86–88.

Wagstaff, K., & Cardie, C. (2000). Clustering with instance-level constraints. *ICML* (pp. 1103–1110).

Wagstaff, K., Cardie, C., Rogers, S., & Schroedl, S. (2001). Constrained k-means clustering with background knowledge. *ICML* (pp. 577–584).

Yu, S. X., & Shi, J. (2001). Grouping with bias. *NIPS*.