Bayesian Active Clustering with Pairwise Constraints

Yuanli Pei(✉), Li-Ping Liu, and Xiaoli Z. Fern

School of EECS, Oregon State University, Corvallis, Oregon 97331, USA
{peiy,liuli,xfern}@eecs.oregonstate.edu

Abstract. Clustering can be improved with *pairwise constraints* that specify similarities between pairs of instances. However, randomly selecting constraints could lead to the waste of labeling effort, or even degrade the clustering performance. Consequently, how to *actively* select effective pairwise constraints to improve clustering becomes an important problem, which is the focus of this paper. In this work, we introduce a Bayesian clustering model that learns from pairwise constraints. With this model, we present an active learning framework that iteratively selects the most informative pair of instances to query an oracle, and updates the model posterior based on the obtained pairwise constraints. We introduce two information-theoretic criteria for selecting informative pairs. One selects the pair with the most uncertainty, and the other chooses the pair that maximizes the marginal information gain about the clustering. Experiments on benchmark datasets demonstrate the effectiveness of the proposed method over state-of-the-art.

1 Introduction

Constraint-based clustering aims to improve clustering using user-provided *pairwise constraints* regarding similarities between pairs of instances. In particular, a must-link constraint states that a pair of instances belong to the same cluster, and a cannot-link constraint implies that two instances are in different clusters. Existing work has shown that such constraints can be effective at improving clustering in many cases [2,4,8,16,19,20,22,24,28]. However, most prior work focus on "passive" learning from constraints, i.e., instance pairs are randomly selected to be labeled by a user. Constraints acquired in this random manner may be redundant and lead to the waste of labeling effort, which is typically limited in real applications. Moreover, when the constraints are not properly selected, they may even be harmful to the clustering performance as has been revealed by Davidson et al. [7]. In this paper, we study the important problem of *actively* selecting effective pairwise constraints for clustering.

Existing work on active learning of pairwise constraints for clustering has mostly focused on neighbourhood-based methods [3,12,14,17,25]. Such methods maintain a neighbourhood structure of the data based on the existing constraints, which represents a partial clustering solution, and they query pairwise constraints to expand such neighborhoods. Other methods that do not rely on

An erratum to this chapter is available at DOI: 10.1007/978-3-319-23528-8_44

A. Appice et al. (Eds.): ECML PKDD 2015, Part I, LNAI 9284, pp. 235–250, 2015.
DOI: 10.1007/978-3-319-23528-8_15

such structure consider various criteria for measuring the utility of instance pairs. For example, Xu et al. [26] propose to select constraints by examining the spectral eigenvectors of the similarity matrix, and identify data points that are at or close to cluster boundaries. Vu et al. [21] introduce a method that chooses instance pairs involving points on the sparse regions of a k-nearest neighbours graph. As mentioned by Xiong et al. [25], many existing methods often select a batch of pairwise constraints before performing clustering, and they are not designed for iteratively improving clustering by querying new pairs.

In this work, we study Bayesian active clustering with pairwise constraints in an iterative fashion. In particular, we introduce a Bayesian clustering model to find the clustering posterior given a set of pairwise constraints. At every iteration, our task is: a) to select the most informative pair toward improving current clustering, and b) to update the clustering posterior after the query is answered by an oracle/a user. Our goal is to achieve the best possible clustering performance with minimum number of queries.

In our Bayesian clustering model, we use a discriminative logistic model to capture the conditional probability of the cluster assignments given the instances. The likelihood of observed pairwise constraints is computed by marginalizing over all possible cluster assignments using message passing. We adopt a special data-dependent prior that encourages large cluster separations. At every iteration, the clustering posterior is represented by a set of samples ("particles"). After obtaining a new constraint, the posterior is effectively updated with a sequential Markov Chain Monte Carlo (MCMC) method ("particle filter").

We present two information-theoretic criteria for selecting instance pairs to query at each iteration: a) *Uncertain*, which chooses the most uncertain pair based on current posterior, and b) *Info*, which selects the pair that maximizes the information gain regarding current clustering. With the clustering posterior maintained at every iteration, both objectives can be efficiently calculated.

We evaluate our method on benchmark datasets, and the results demonstrate that our Bayesian clustering model is very effective at learning from a small number of pairwise constraints, and our active clustering model outperforms state-of-the-art active clustering methods.

2 Problem Statement

The goal of clustering is to find the underlying cluster structure in a dataset $X = [x_1, \cdots, x_N]$ with $x_i \in \mathbb{R}^d$ where d is the feature dimension. The unknown cluster label vector $Y = [y_1, \cdots, y_N]$, with $y_i \in \{1, \cdots, K\}$ being the cluster label for x_i, denotes the ideal clustering of the dataset, where K is the number of clusters. In the studied *active* clustering, we could acquire some weak supervision, i.e., pairwise constraints, by requesting an oracle to specify whether two instances $(x_a, x_b) \in X \times X$ belong to the same cluster. We represent the response of the oracle as a pair label $z_{a,b} \in \{+1, -1\}$, with $z_{a,b} = +1$ representing that instance x_a and x_b are in the same cluster (a must-link constraint), and $z_{a,b} = -1$ meaning that they are in different clusters (a cannot-link constraint). We assume the cost

is uniform for different queries, and the goal of active clustering is to achieve the best possible clustering with the least number of queries.

In this work, we consider sequential active clustering. In each iteration, we select one instance pair to query the oracle. After getting the answer of the query, we update the clustering model to integrate the supervision. With the updated model, we then choose the best possible pair for the next query. So the task of active clustering is an iterative process of posing queries and incorporating new information to clustering.

An active clustering model generally has two key components: the *clustering* component and the *pair selection* component. In every iteration, the task of the clustering component is to identify the cluster structure of the data given the existing constraints. The task of the pair selection component is to score each candidate pair and choose the most informative pair to improve the clustering.

3 Bayesian Active Clustering

3.1 The Bayesian Clustering Model

In our model, we assume that the instance cluster labels y_i's are independent given instance x_i and the model parameter W. Each pair label $z_{a,b}$ only depends on the cluster labels y_a and y_b of the involved instances (x_a, x_b). The proposed Bayesian clustering model consists of three elements: 1) the instance cluster assignment model defined by $P(Y|W, X)$, with parameter W; 2) the conditional distribution of the pair labels given the cluster labels $P(Z|Y)$, where Z contains all pair labels in the constraints; and 3) the data-dependent prior $P(W|X, \theta)$ with parameter θ. The joint distribution of the clustering model is factorized as

$$P(Z, Y, W|X, \theta) = P(Z|Y)P(Y|W, X)P(W|X, \theta) \ . \tag{1}$$

We use the following discriminative logistic model as the clustering assignment model $P(Y|W, X)$:

$$P(y_i = k|W, x_i) = \frac{\exp(W_{\cdot,k}^\top x_i)}{\sum_{k'=1}^{K} \exp(W_{\cdot,k'}^\top x_i)}, \quad \forall 1 \le k \le K, \quad 1 \le i \le N \ , \tag{2}$$

where W is a $d \times K$ matrix, d is the feature dimension, and K is the number of clusters.

Here we use a special prior for W, which combines the Gaussian prior with a data-dependent term that encourages large cluster separations of the data. The logarithmic form of the prior distribution is

$$\log \ P(W|X, \theta) = -\frac{\lambda}{2}\|W\|_F^2 - \frac{\tau}{N}\sum_{i=1}^{N} H(y_i|W, x_i) + constant \ , \tag{3}$$

where the prior parameter $\theta = [\lambda, \tau]$. The first term is the weighted Frobenius norm of W. This term corresponds to the Gaussian prior with zero mean and

diagonal covariance matrix with λ as the diagonal elements, and it controls the model complexity. The second term is the average negative entropy of the cluster assignment variable Y. We use this term to encourage large separations among clusters, as similarly utilized by [11] for semi-supervised classification problems. The constant term normalizes the probability. Although it is unknown, inference can be carried out by sampling from the unnormalized distribution (e.g., using slice sampling [18]). We will discuss more details in Sec. 3.3.

With our model assumption, the conditional probability $P(Z|Y)$ is fully factorized based on the pairwise constraints. For a single pair (x_a, x_b), we define the probability of $z_{a,b}$ given cluster labels y_a and y_b as

$$\begin{aligned} P(z_{a,b} = +1|y_a, y_b) &= \begin{cases} \epsilon & \text{if } y_a \neq y_b \\ 1-\epsilon & \text{if } y_a = y_b \end{cases} , \\ P(z_{a,b} = -1|y_a, y_b) &= 1 - P(z_{a,b} = +1|y_a, y_b) \ , \end{aligned} \tag{4}$$

where ϵ is a small number to accommodate the (possible) labeling error. In the case where no labeling error exists, ϵ allows for "soft constraints", meaning that the model can make small errors on some pair labels and achieve large cluster separations.

Marginalization of Cluster Labels. In the learning procedure described later, we will need to marginalize some or all cluster labels, for example, in the case of computing the likelihood of the observed pair labels:

$$P(Z|W, X) = \sum_{Y} P(Z, Y|W, X) = \sum_{Y_{\alpha(Z)}} P(Z|Y_{\alpha(Z)}) P(Y_{\alpha(Z)}|W, X_{\alpha(Z)}) \ , \tag{5}$$

where $\alpha(Z)$ denotes the set of indices for all instances involved in Z.

The marginalization can be solved by performing sum-product message passing [15] on a factor graph defined by all the constraints. Specifically, the set of all instances indexed by $\alpha(Z)$ defines the nodes of the graph, and $P(Y_{\alpha(Z)}|W, X_{\alpha(Z)})$ defines the node potentials. Each queried pair (x_a, x_b) creates an edge, and the edge potential is defined by $P(z_{a,b}|y_a, y_b)$. In this work, we require that the graph formed by the constraints does not contain cycles, and message passing is performed on a tree (or a forest, which is a collection of trees). Since inference on trees are exact, the marginalization is computed exactly. Moreover, due to the simple form of the edge potential (which is a simple modification to the identity matrix as can be seen from (4)), the message passing can be performed very efficiently. In fact, each message propagation only requires $O(K)$ complexity instead of $O(K^2)$ as in the general case. Overall the message passing only takes $O(K|Z|)$, even faster than calculating the node potentials $P(Y_{\alpha(Z)}|W, X_{\alpha(Z)})$, which takes $O(dK|Z|)$.

3.2 Active Query Selection

Now we describe our approach for actively selecting informative pairs at every iteration. Suppose our query budget is T. In each iteration $t, 1 \leq t \leq T$, we need

to select a pair (x_a^t, x_b^t) from a pool of unlabeled pairs U^t, and acquire the label $z_{a,b}^t$ from the oracle. We let $U^1 \subseteq X \times X$ be the initial pool of unlabeled pairs. Then $U^t = U^{t-1} \backslash (x_a^{t-1}, x_b^{t-1})$ for $1 \leq t \leq T$. Below we use $Z_t = [z_{a,b}^1, \cdots, z_{a,b}^t]$ to denote all the pair labels obtained up to the t-th iteration.

Selection Criteria. We use two entropy-based criteria to select the best pair at each iteration. The first criterion, which we call *Uncertain*, is to select the pair whose label is the most uncertain. That is, at the t iteration, we choose the pair (x_a^t, x_b^t) that has the largest *marginal* entropy of $z_{a,b}^t$ (over the posterior distribution of W):

$$(x_a^t, x_b^t) = \arg \max_{(x_a, x_b) \in U^t} H(z_{a,b} | Z_{t-1}, X, \theta) \ . \tag{6}$$

Similar objective has been considered in prior work on distance metric learning [27] or document clustering [14], where the authors propose different approaches to compute/approximate the entropy objective.

The second criterion is a greedy objective adopted from active learning for classification [6,10,13], which we call *Info*. The idea is to select the query (x_a^t, x_b^t) that maximizes the marginal information gain about the model W:

$$\begin{aligned} (x_a^t, x_b^t) &= \arg \max_{(x_a, x_b) \in U^t} I(z_{a,b}, W | Z_{t-1}, X, \theta) \\ &= \arg \max_{(x_a, x_b) \in U^t} H(z_{a,b} | Z_{t-1}, X, \theta) - H(z_{a,b} | W, Z_{t-1}, X, \theta) \ . \end{aligned} \tag{7}$$

Note that here W is a random variable. The *Info* objective is equivalent to maximizing the entropy reduction about W, as can be proved by the chain rule of conditional entropy.

Interestingly, the first entropy term in the *Info* objective (7) is the same with the *Uncertain* objective (6). The additional term to *Info* is the conditional entropy of the pair label $z_{a,b}$ given W, i.e., the second term in (7). Comparing the two objectives, we see that W is marginalized in the *Uncertain* objective and the selected query aims to reduce the maximum uncertainty of the *pair label*. In contrast, the goal of *Info* is to decrease the *model* uncertainty. There is subtle difference between these two types of uncertainties. The additional conditional entropy term in *Info* suggests that it prefers instance pairs whose labels are certain once W is known, yet whose overall uncertainty is high when marginalizing over W. In such sense, *Info* pays more attention to the uncertainty of the model W.

Each of the above selection objectives ranks the candidate pairs from the highest to the lowest. To select a pair to query, we go through the ranking and choose the one that does not create a cycle to the existing graph as described in Sec. 3.1. Since inference on trees are not only exact but also fast, enforcing such acyclic graph structure allows us to compute the selection objectives more effectively and accurately, and select more informative pairs to query.

Computing the Selection Objectives. Now we describe how to compute the two objective values for a candidate instance pair. The two objectives require computing the marginal entropy $H(z_{a,b}|Z_t, X, \theta)$, and the conditional entropy $H(z_{a,b}|W, Z_t, X, \theta)$, for $1 \leq t \leq T$. By definition, the marginal entropy is

$$H(z_{a,b}|Z_t, X, \theta) = -\sum_{z_{a,b}} P(z_{a,b}|Z_t, X, \theta) \log P(z_{a,b}|Z_t, X, \theta) \ , \tag{8}$$

where the probability

$$P(z_{a,b}|Z_t, X, \theta) = \int P(\hat{W}|Z_t, X, \theta) P(z_{a,b}|Z_t, \hat{W}, X) d\hat{W} \ . \tag{9}$$

The conditional probability is computed as

$$P(z_{a,b}|Z_t, \hat{W}, X) = \frac{P(z_{a,b} \cup Z_t|\hat{W}, X)}{P(Z_t|\hat{W}, X)} \ , \tag{10}$$

where calculating both the numerator and the denominator are the same inference problem as (5) and can be solved similarly using message passing. In fact, message propagations for the two calculations are shared except for that a new edge regarding $z_{a,b}$ is introduced to the graph for $P(z_{a,b} \cup Z_t|\hat{W}, X)$. So we can calculate the two values by performing message passing algorithm only once on the graph of $P(z_{a,b} \cup Z_t|\hat{W}, X)$, and record $P(Z_t|\hat{W}, X)$ in the intermediate step.

By definition, the conditional entropy is

$$H(z_{a,b}|W, Z_t, X, \theta) = \int P(\hat{W}|Z_t, X, \theta) H(z_{a,b}|Z_t, \hat{W}, X) d\hat{W} \ , \tag{11}$$

where $H(z_{a,b}|\hat{W}, Z_t, X)$ is also easy to compute once we know $P(z_{a,b}|Z_t, \hat{W}, X)$, which has been done in (10).

Now the only obstacle in calculating the two entropies is to take the expectations over the posterior distribution $P(W|Z_t, X, \theta)$ in (9) and (11). Here we use sampling to approximate such expectations. We first sample W's from $P(W|Z_t, X, \theta)$ and then approximate the expectations with the sample means. Directly sampling from the posterior at every iteration is doable but very inefficient. Below we describe a sequential MCMC sampling method ("particle filter") that effectively updates the samples of the posterior.

3.3 The Sequential MCMC Sampling of W

The main idea of the sequential MCMC method is to avoid sampling with random starts at every iteration by utilizing the particles obtained from the previous iteration.[1] Specifically, to obtain particles from distribution $P(W|Z_t, X, \theta)$, the sequential MCMC method first resamples from the particles previously sampled

[1] Here we follow the convention of the particle filter field and call samples of W as "particles".

from $P(W|Z_{t-1}, X, \theta)$, and then performs just a few MCMC steps with these particles to prevent degeneration [9].

Here we maintain S particles in each iteration. We denote W_s^t, $1 \leq s \leq S$, as the s-th particle in the t-th iteration. For initialization, we sample particles $\{W_1^0, \cdots, W_S^0\}$ from the prior distribution $P(W|X, \theta)$ defined in (3) using slice sampling [18] [2], an MCMC method that can uniformly draw samples from an unnormalized density function. Since slice sampling does not require the target distribution to be normalized, the unknown constant in the prior (3) can be neglected here.

At iteration $t, 1 \leq t \leq T$, after a new pair label $z_{a,b}^t$ is observed, we perform the following two steps to update the particles and get samples from $P(W|Z_t, X, \theta)$.

(1) Resample. The first step is to resample from the particles $\{W_1^{t-1}, \cdots, W_S^{t-1}\}$ obtained from the previous iteration for $P(W|Z_{t-1}, X, \theta)$. We observe that

$$\begin{aligned} P(W|Z_t, X, \theta) &= P(W|z_{a,b}^t, Z_{t-1}, X, \theta) \\ &\propto P(z_{a,b}^t|Z_{t-1}, W, X)P(W|Z_{t-1}, X, \theta) \ . \end{aligned}$$

So each particle W_s^{t-1} is weighted by $P(z_{a,b}^t|Z_{t-1}, W_s^{t-1}, X)$, which can be calculated the same as (10).

(2) Move. In the second step, we start with each resampled particles, and perform several slice sampling steps for the posterior

$$P(W|Z_t, X, \theta) \propto P(Z_t|W, X)P(W|X, \theta) \ . \tag{12}$$

Again $P(Z_t|W, X)$ is calculated by message passing as (5), and the unknown normalizing constant in $P(W|X, \theta)$ can be ignored, since slice sampling does not require the normalization constant.

The *resample-move* method avoids degeneration in the sequence of slice sampling steps. After these two steps, we have updated the particles for $P(W|Z_t, X, \theta)$. Such particles are used to approximate the selection objectives as described in Sec. 3.2, allowing us to select the next informative pair to query.

Note that the distribution $P(W|Z_t, X, \theta)$ is invariant to *label switching*, that is, permuting column vectors of $W = [W_{\cdot,1}, \cdots, W_{\cdot,K}]$ will not change the probability $P(W|Z_t, X, \theta)$. This is because we can not provide any prior of W with label order, nor does the obtained constraints provide any information about the label order. One concern is whether the label switching problem would reduce sampling efficiency and affect the pair selection, since $P(W|Z_t, X, \theta)$ has multiple modes corresponding to different label permutations. Actually it does not cause an issue to the approximation of integrations in (9) and (11), since the term $P(z_{a,b}|Z_t, W, X, \theta)$ is also invariant to label permutations. However, the label switching problem does cause difficulty in getting the Bayesian prediction of clusters labels from distribution $P(Y|Z_t, X, \theta)$, so we will employ the MAP solution W_{map} and predict cluster labels with $P(Y|Z_t, W_{map}, X, \theta)$. We describe this in the following section.

[2] Here we use the implementation `slicesample` provided in the MATLAB toolbox.

3.4 Find the MAP Solution

Given a set of constraints with pair labels Z, we first find the MAP estimation W_{map} by maximizing the posterior $P(W|Z, X, \theta)$, or equivalently maximizing the joint distribution $P(W, Z|X, \theta)$ (in the logarithmic form):

$$\max_{W} \quad L \;=\; \log P(W, Z|X, \theta) \;=\; \log P(Z|W, X) + \log P(W|X, \theta) \;. \qquad (13)$$

The maximization can be solved by off-the-shelf gradient-based optimization approaches. Here we use the quasi-newton method provided in the MATLAB toolbox. The gradient of the objective L with respect to W is

$$\frac{\partial L}{\partial W} = \sum_{i \in \alpha(Z)} x_i (q_i - p_i)^\top - \lambda W - \frac{\tau}{N} \sum_{i=1}^{N} x_i \sum_{k=1}^{K} p_{ik} \log p_{ik} (\mathbf{1}_k - p_i)^\top \;,$$

where $p_i = [p_{i1}, \cdots, p_{iK}]^\top$ with $p_{ik} = P(y_i = k|W, x_i)$, $q_i = [q_{i1}, \cdots, q_{iK}]^\top$ with $q_{ik} = P(y_i = k|Z, W, x_i)$, and $\mathbf{1}_k$ is a K dimensional vector that contains 1 on the k-th dimension and 0 elsewhere. Here $\alpha(Z)$ again indexes all the instances involved in the constraints.

With the W_{map} solution to (13), we then find the MAP solution of the cluster labels Y from $P(Y|Z, W_{map}, X)$. This is done in two cases. For the instances that are *not* involved in the constraints, the MAP of Y is simply the most possible assignment of $P(Y|W_{map}, X)$. For the instances involved in the constraints, we need to find

$$\max_{Y_{\alpha(Z)}} \quad P(Y_{\alpha(Z)}|Z, W_{map}, X_{\alpha(Z)}) \propto P(Z|Y_{\alpha(Z)}) P(Y_{\alpha(Z)}|W_{map}, X_{\alpha(Z)}) \;.$$

The inference can be done by performing max-product algorithm on the same graph as defined for (5), only replacing the "summation" with the "max" operator at every message propagation.

In real applications, we only need to find the MAP solution of Y after the last iteration. In our experiments, we search for the solution at every iteration to show the performance of our method if we stop learning at any iteration. Our overall algorithm is summarized in Algorithm 1.

Note that an alternative of finding the clustering solution is to find the MAP of W and Y at the same time. However, we think our MAP estimation of W which marginalizes Y is more stable, and our calculation method is much simpler compared with the alternative.

4 Experiments

In this section, we empirically examine the effectiveness of the proposed method. In particular, we aim to answer the following questions:

- Is the proposed Bayesian clustering model effective at finding good clustering solutions with a small number of pairwise constraints?
- Is the proposed *active* clustering method more effective than state-of-the-art active clustering approaches?

Algorithm 1. Bayesian Active Clustering

Input: data X, number of clusters K, access to the oracle, initial pool U^1, query budget T, prior parameter θ, number of samples S
Output: a clustering solution of the data

Initialize particles by sampling $\{W_1^0, \cdots, W_S^0\}$ from prior $P(W|X,\theta)$
for $t = 1$ **to** T **do**
1. Select a pair to query:
Use particles $\{W_1^{t-1}, \cdots, W_S^{t-1}\}$ to compute the selection objective (6) or (7)
Choose the best pair (x_a^t, x_b^t) from U^t and acquire $z_{a,b}^t$ from the oracle
2. Update posterior:
Resample S particles with weight $P(z_{a,b}^t|Z_{t-1}, W_s^{t-1}, X)$ for W_s^{t-1}
Perform a few MCMC steps on all particles with distribution $P(W|Z_t, X, \theta)$
3. Update the pool: $U^{t+1} \leftarrow U^t \backslash (x_a^t, x_b^t)$
end for
Find the MAP solution $W_{map} = \arg\max_W \log P(W|Z_T, X, \theta)$
Find the clustering solution $Y_{map} = \arg\max_Y \log P(Y|Z_T, W_{map}, X)$

Table 1. Summary of Dataset Information

Dataset	#Inst	#Dim	#Class	#Query
Fertility	100	9	2	60
Parkinsons	195	22	2	60
Crabs	200	5	2	60
Sonar	208	60	2	100
Balance	625	4	3	100
Transfusion	748	4	2	100
Letters-IJ	1502	16	2	100
Digits-389	3165	16	3	100

4.1 Dataset and Setup

We use 8 benchmark UCI datasets to evaluate our method. Table 1 provides a summary of the dataset information. For each dataset, we normalize all features to have zero mean and unit standard deviation.

We form the pool of unlabeled pairs using all instances in the dataset, and set the query budget to 60 for smaller datasets and to 100 for datasets with large feature dimension (e.g, *Sonar*) or larger dataset size. When a pair of instances is queried, the label is returned based on the ground-truth instance class/cluster labels. We evaluate the clustering results of all methods using pairwise F-Measure [5], which evaluates the harmonic mean of the precision and recall regarding prediction of instance pairwise relations. We repeat all experiments 30 times and average the results.

For the proposed Bayesian clustering model, we found that its performance is not sensitive to the values of the prior parameter τ or the ϵ used in the pair label distribution (4). Here we set $\tau = 1$ and $\epsilon = 0.05$, where the nonzero value of ϵ

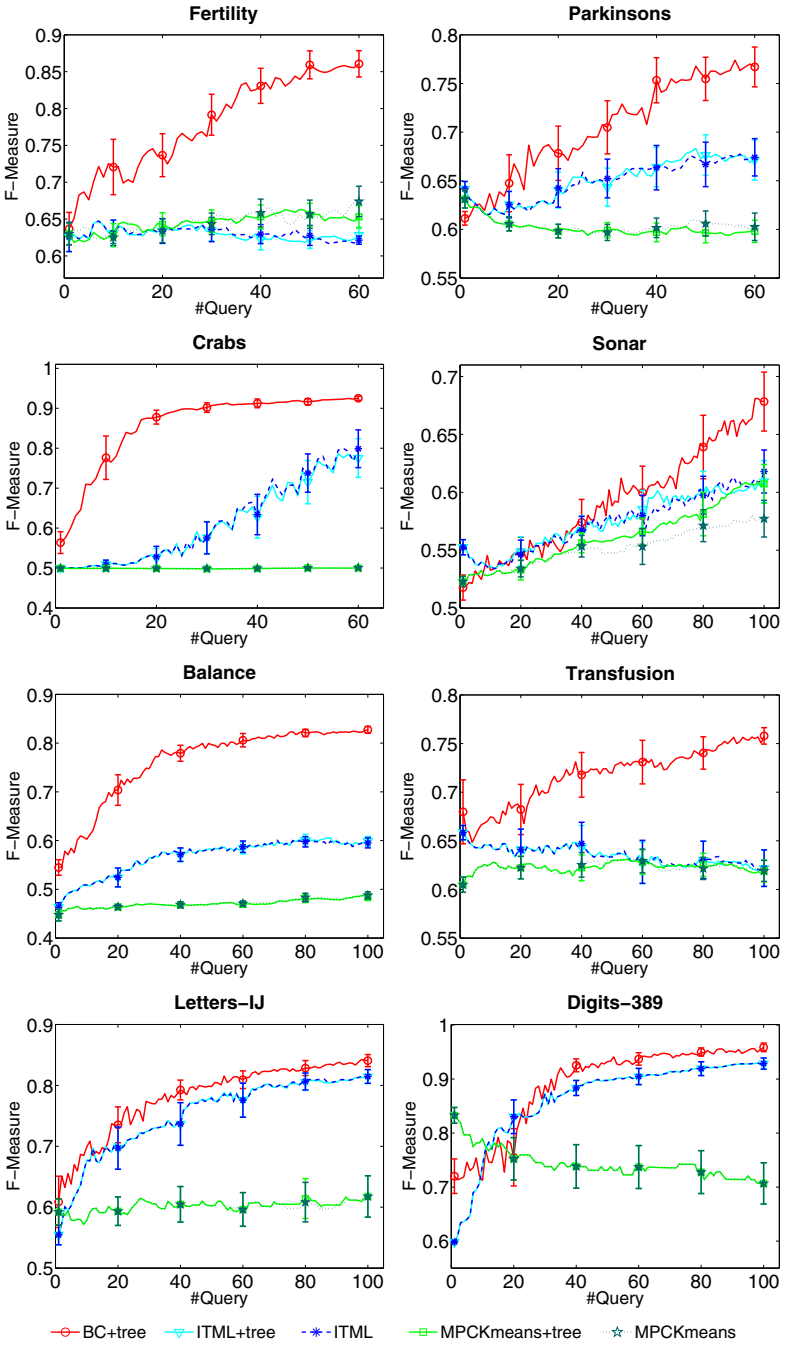

Fig. 1. Pairwise F-Measure clustering results with increasing number of *randomly* selected queries. Results are averaged over 30 runs. Error bars are shown as mean and 95% confidence interval.

allows for "soft constraints". For the parameter λ, which controls the covariance of the Gaussian prior, we experimented with $\lambda \in \{1, 10, 100\}$ and found that $\lambda = 10$ is uniformly good with all datasets, which we fix as the default value. For each dataset, we maintain $S = 2dK$ samples of the posterior at every iteration.

4.2 Effectiveness of the Proposed Clustering Model

To demonstrate the effectiveness of the proposed Bayesian clustering (BC) model, we compare with two well-known methods that learn from pairwise constraints: MPCKmeans [5], and ITML [8][3]. In this set of experiment, we use randomly selected pairwise constraints to evaluate all methods. For our method, we incrementally select random pairs that do not introduce a cycle to the graph formed by existing pairs. To ensure a fair comparison, we evaluate ITML and MPCKmeans with randomly selected pairs with and without the acyclic graph restriction. Thus, all methods in competition are: *BC+tree*, *ITML*, *ITML+tree*, *MPCKmeans*, *MPCKmeans+tree*, where *BC+tree*, *ITML+tree*, and *MPCKmeans+tree* use randomly selected constraints that form a tree graph (or a forest), and *ITML* and *MPCKmeans* allow for cycles in the graph.

Figure 1 shows the performance of all methods with increasing number of constraints. We see that our method *BC+tree* outperforms the baselines on most datasets regardless of whether they use constraints with or without the acyclic graph restriction. This demonstrates the effectiveness of our Bayesian clustering model. We also notice that on most datasets we can hardly tell the difference between *ITML* and *ITML+tree*, or *MPCKmeans* and *MPCKmeans+tree*, suggesting that enforcing the acyclic structure in the constraints do not hurt the performance of ITML or MPCKmeans. Interestingly, such enforcement can in some cases produce better performance (e.g, on the *Sonar* dataset). We suspect this is because constraints forming cycles may have larger *incoherence* than those does not.[4] Davidson et al. [7] have shown that constraint sets with large incoherence can potentially degrade the clustering performance.

4.3 Effectiveness of the Overall Active Clustering Model

In this section, we compare our overall active clustering model with existing methods. Our baselines include two recent work on active learning with pairwise constraints: *MinMax* [17], and *NPU* [25]. Both methods provide an active pair selection approach and require a clustering method to learn form the constraints. Here we supply them with MPCKmeans and ITML.[5] So all methods in competition are

[3] ITML is a distance metric learning method, and we find the clustering solution by applying Kmeans clustering with the learned metric.

[4] The concept of *incoherence* is formally defined at [7]. Generally, a set of overlapping constraints tends to have higher incoherence than a set of disjoint constraints.

[5] Note that due to our Bayesian clustering model requires the set of constraints to form an acyclic graph, it can not be combined with *MinMax* or *NPU*, as they generally select constraints that form cycles due to their neighbourhood-based approach.

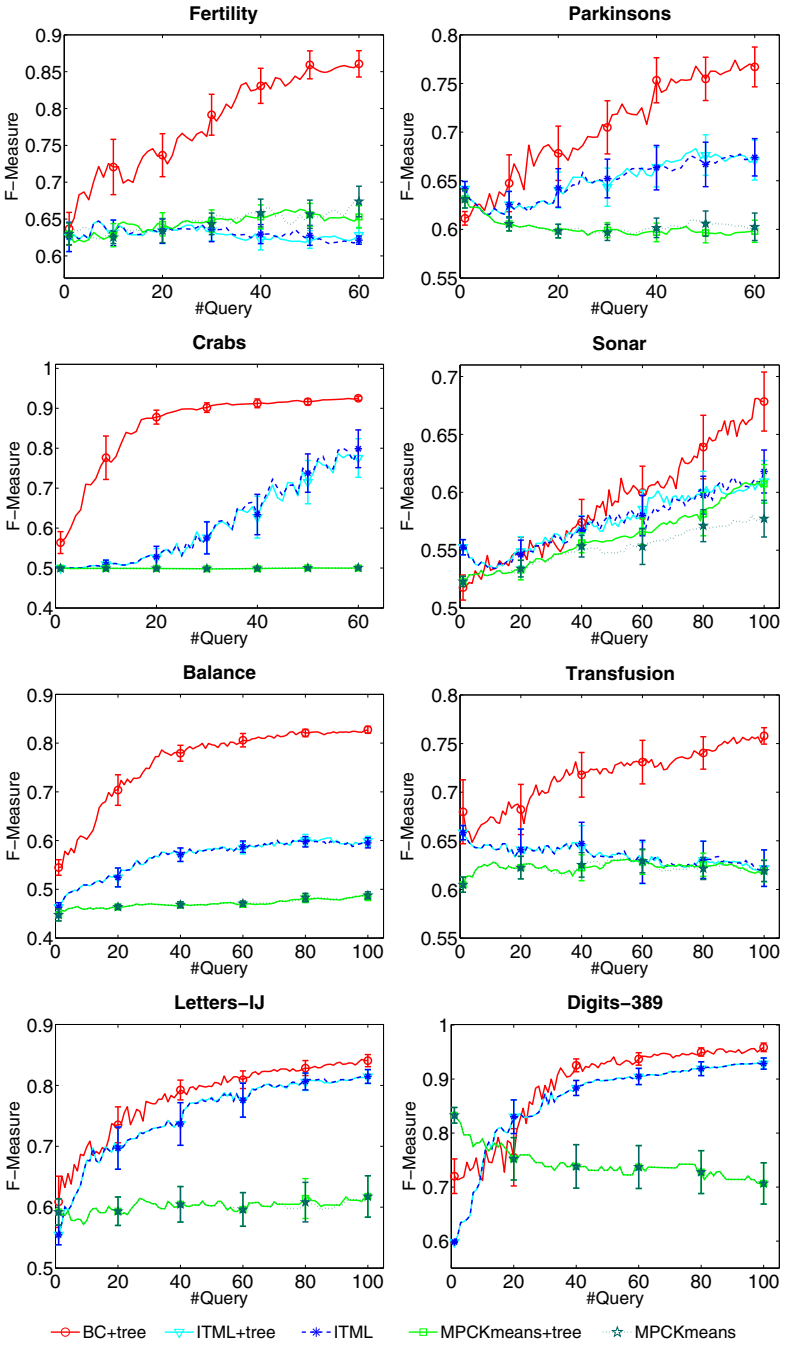

Fig. 2. Pairwise F-Measure clustering results of different *active clustering* methods with increasing number of queries. Results are averaged over 30 runs. Error bars are shown as mean and 95% confidence interval.

- *Info+BC*: The proposed active clustering model with the *Info* criterion (7).
- *Uncertain+BC*: The proposed active clustering model with the *Uncertain* criterion (6).
- *NPU+ITML*: The *NPU* active selection strategy combined with *ITML*.
- *NPU+MPCKmeans*: The *NPU* method with *MPCKmeans*.
- *MinMax+ITML*: The *MinMax* active learning method combined with *ITML*.
- *MinMax+MPCKmeans*: The *MinMax* approach combined with *MPCKmeans*.

Figure 2 reports the performance of all active clustering methods with increasing number of queries. We see that both *Info+BC* and *Uncertain+BC* improve the clustering very quickly as more constraints are obtained, and they outperform all baselines on most datasets. Moreover, *Info+BC* seems to be more effective than *Uncertain+BC* in most cases. We hypothesize this is because *Info* reduces the uncertainty of the model, which might be more appropriate for improving the MAP solution of clustering than decreasing the maximum uncertainty of the pair labels as *Uncertain* does.

To avoid crowding Fig. 2, we did not present the passive learning results of our method *BC+tree* as a baseline in the same figure. The comparison between active learning and passive learning for our method can be done by comparing *Uncertain+BC* and *Info+BC* in Fig. 2 with *BC+tree* in Fig. 1. We see that both our active learning approaches produce better performance than passive learning on most datasets, demonstrating the effectiveness of our pair selection strategies.

We also notice that the performance of *NPU* or *MinMax* highly depends on the clustering method in use. With different clustering methods, their behaviors are very different. In practice, it can be difficult to decide which clustering algorithm should be used in combination with the active selection strategies to ensure good clustering performance. In contrast, our method unifies the clustering and active pair selection model, and the constraints are selected to explicitly reduce the clustering uncertainty and improve the clustering performance.

4.4 Analysis of the Acyclic Graph Restriction

Our method requires the graph formed by the constraints to be a tree (or a forest). Here we show that this restriction will not prevent us from selecting informative pairs. We examine the number of pairs that has been dropped at every iteration in order to find the best pair that does not create a cycle. Table 2 reports the results for the two selection criteria with varied number of queries. We see that for both criteria the number of dropped pairs is very small. For *Uncertain*, there is barely any pair that has been dropped on most datasets, and we see slightly more pairs dropped for the *Info* criteria. Overall, for only less than (often significantly less than) 10% of the number of queries, we encounter the need of dropping a pair. The only exception is the *Fertility* dataset, which is very small in size, making it difficult to avoid cycles with a large number of queries. But from the results in Sec. 4.3, we can see that the active clustering performance was still much better than the competing methods.

Table 2. Number of dropped pairs (*Info*/*Uncertain*) at different iterations to find the best pair that does not a create cycle. Results are averaged over 30 runs.

Dataset	Query Iteration					
	10	20	30	40	50	60
Fertility	0.4/0.0	0.6/0.1	0.9/0.1	2.7/1.9	4.2/14.3	10.8/32.0
Parkinsons	0.1/0.0	0.0/0.0	0.5/0.0	0.8/0.3	0.9/0.6	1.7/1.7
Crabs	0.6/0.0	0.2/0.0	0.0/0.0	0.1/0.3	0.2/0.6	0.4/1.5
Sonar	0.7/0.0	0.2/0.0	0.4/0.1	0.5/0.2	0.5/0.2	0.6/0.2
Balance	0.0/0.0	0.3/0.0	1.7/0.0	2.6/0.0	3.3/0.1	2.9/0.0
Transfusion	0.3/0.0	1.3/0.0	2.4/0.0	2.3/0.0	4.6/0.0	4.9/0.1
Letters-IJ	0.0/0.0	0.2/0.0	0.3/0.0	0.2/0.0	0.5/0.0	0.7/0.0
Digits-389	0.0/0.0	0.0/0.0	0.1/0.0	0.1/0.0	0.0/0.0	0.3/0.0

In addition, during our experiments, we found that for both criteria the difference between the maximum objective value and objective of the finally selected pair is often negligible. So in the case where some high-ranking pairs are dropped due to the acyclic graph structure restriction, the selected pair is still very informative. Overall, this enforcement does not present any significant negative impact on the final clustering results. It is interesting to note that, the results in Sec. 4.2 suggest that such graph structure restriction can in some cases improve the clustering performance.

5 Related Work

Prior work on active clustering for pairwise constraints has mostly focused on the neighbourhood-based method, where a neighbourhood skeleton is constructed to partially represent the underlying clusters, and constraints are queried to expand such neighbourhoods. Basu et al. [3] first proposed a two-phase method, Explore and Consolidate. The Explore phase incrementally builds K disjoint neighborhoods by querying instance pairwise relations, and the Consolidate phase iteratively queries random points outside the neighborhoods against the existing neighborhoods, until a must-link constraint is found. Mallapragada et al. [17] proposed an improved version, which modifies the Consolidate stage to query the most uncertain points using an MinMax objective. As mentioned by Xiong et al. [25], these methods often select a batch of constraints before performing clustering, and they are not designed for iteratively improving clustering by querying new constraints, as considered in this work.

Wang and Davidson [23], Huang et al. [14] and Xiong et. al. [25] studied active clustering in an iterative manner. Wang and Davidson introduced an active spectral clustering method that iteratively select the pair that maximized the expected error reduction of current model. This method is however restricted to the two-cluster problems. Huang et al. proposed an active document clustering method that iteratively finds probabilistic clustering solution using a language model and they selected the most uncertain pair to query. But this method is

limited to the task of document clustering. Xiong et. al. considered a similar iterative framework to Huang et al., and they queried the most uncertain data point against existing neighbourhoods, as apposed to the most uncertain pair in [14]. Xiong et al. only provide a query selection strategy and require a clustering method to learn from the constraints. In contrast, our method is a unified clustering and active pair selection model.

Finally, there are other methods that use various criteria to select pairwise constraints. Xu et al. [26] proposed to select constraints by examining the spectral eigenvectors of the similarity matrix in the two-cluster scenario. Vu et al. [21] proposed to select constraints involving points on the sparse regions of a k-nearest neighbours graph. The work [1,12] used ensemble approaches to select constraints. The scenarios considered in these methods are less similar to what has been studied in this paper.

6 Conclusion

In this work, we studied the problem of active clustering, where the goal is to iteratively improve clustering by querying informative pairwise constraints. We introduced a Bayesian clustering method that adopted a logistic clustering model and a data-dependent prior which controls model complexity and encourages large separations among clusters. Instead of directly computing the posterior of the clustering model at every iteration, our approach maintains a set of samples from the posterior. We presented a sequential MCMC method to efficiently update the posterior samples after obtaining a new pairwise constraint. We introduced two information-theoretic criteria to select the most informative pairs to query at every iteration. Experimental results demonstrated the effectiveness of the proposed Bayesian active clustering method over existing approaches.

Acknowledgments. This material is based upon work supported by the National Science Foundation under Grant No. IIS-1055113. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Al-Razgan, M., Domeniconi, C.: Clustering ensembles with active constraints. In: Okun, O., Valentini, G. (eds.) Applications of Supervised and Unsupervised Ensemble Methods. SCI, vol. 245, pp. 175–189. Springer, Heidelberg (2009)
2. Baghshah, M.S., Shouraki, S.B.: Semi-supervised metric learning using pairwise constraints. In: IJCAI, pp. 1217–1222 (2009)
3. Basu, S., Banerjee, A., Mooney, R.J.: Active semi-supervision for pairwise constrained clustering. In: SDM, pp. 333–344 (2004)
4. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: KDD, pp. 59–68 (2004)
5. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: ICML, pp. 81–88 (2004)

6. Dasgupta, S.: Analysis of a greedy active learning strategy. In: NIPS, pp. 337–344 (2005)
7. Davidson, I., Wagstaff, K.L., Basu, S.: Measuring constraint-set utility for partitional clustering algorithms. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 115–126. Springer, Heidelberg (2006)
8. Davis, J.V., Kulis, B., Jain, P., Sra, S., Dhillon, I.S.: Information-theoretic metric learning. In: ICML, pp. 209–216 (2007)
9. Gilks, W.R., Berzuini, C.: Following a moving target - monte carlo inference for dynamic bayesian models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **63**(1), 127–146 (2001)
10. Golovin, D., Krause, A., Ray, D.: Near-optimal bayesian active learning with noisy observations. In: NIPS, pp. 766–774 (2010)
11. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: NIPS, pp. 281–296 (2005)
12. Greene, D., Cunningham, P.: Constraint selection by committee: an ensemble approach to identifying informative constraints for semi-supervised clustering. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 140–151. Springer, Heidelberg (2007)
13. Houlsby, N., Huszár, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning. CoRR abs/1112.5745 (2011)
14. Huang, R., Lam, W.: Semi-supervised document clustering via active learning with pairwise constraints. In: ICDM, pp. 517–522 (2007)
15. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. Inform. Theory **47**(2), 498–519 (2001)
16. Lu, Z., Leen, T.K.: Semi-supervised clustering with pairwise constraints: a discriminative approach. In: AISTATS, pp. 299–306 (2007)
17. Mallapragada, P.K., Jin, R., Jain, A.K.: Active query selection for semi-supervised clustering. In: ICPR, pp. 1–4 (2008)
18. Neal, R.M.: Slice sampling. Annals of statistics **31**(3), 705–741 (2003)
19. Nelson, B., Cohen, I.: Revisiting probabilistic models for clustering with pairwise constraints. In: ICML, pp. 673–680 (2007)
20. Shental, N., Bar-hillel, A., Hertz, T., Weinshall, D.: Computing gaussian mixture models with em using equivalence constraints. In: NIPS, pp. 465–472 (2003)
21. Vu, V.V., Labroche, N., Bouchon-Meunier, B.: An efficient active constraint selection algorithm for clustering. In: ICPR, pp. 2969–2972 (2010)
22. Wagstaff, K.L., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: ICML, pp. 577–584 (2001)
23. Wang, X., Davidson, I.: Active spectral clustering. In: ICDM, pp. 561–568 (2010)
24. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.J.: Distance metric learning with application to clustering with side-information. In: NIPS, pp. 505–512 (2002)
25. Xiong, S., Azimi, J., Fern, X.Z.: Active learning of constraints for semi-supervised clustering. IEEE Trans. Knowl. Data Eng. **26**(1), 43–54 (2014)
26. Xu, Q., desJardins, M., Wagstaff, K.L.: Active constrained clustering by examining spectral eigenvectors. In: Hoffmann, A., Motoda, H., Scheffer, T. (eds.) DS 2005. LNCS (LNAI), vol. 3735, pp. 294–307. Springer, Heidelberg (2005)
27. Yang, L., Jin, R., Sukthankar, R.: Bayesian active distance metric learning. In: UAI, pp. 442–449 (2007)
28. Yu, S.X., Shi, J.: Segmentation given partial grouping constraints. IEEE Trans. Pattern Anal. Mach. Intell. **26**(2), 173–183 (2004)