DISCRIMINATIVE CLUSTERING WITH CARDINALITY CONSTRAINTS

Anh T. Pham, Raviv Raich, and Xiaoli Z. Fern

School of EECS, Oregon State University, Corvallis, OR 97331-5501
{phaman,raich,xfern}@eecs.oregonstate.edu

ABSTRACT

Clustering is widely used for exploratory data analysis in a variety of applications. Traditionally clustering is studied as an unsupervised task where no human inputs are provided. A recent trend in clustering is to leverage user provided side information to better infer the clustering structure in data. In this paper, we propose a probabilistic graphical model that allows user to provide as input the desired cluster sizes, namely the cardinality constraints. Our model also incorporates a flexible mechanism to inject control of the crispness of the clusters. Experiments on synthetic and real data demonstrate the effectiveness of the proposed method in learning with cardinality constraints in comparison with the current state-of-the-art.

1. INTRODUCTION

Clustering is an important task in machine learning, where the goal is to group instances into clusters (categories). There are many practical applications of clustering such as image categorization, image segmentation, document categorization, social network grouping, and bio-informatics. Numerous clustering algorithms have also been proposed in the literature including K-means [1], spectral clustering [2], density based clustering [3], hierarchical clustering [1] and maximum margin based [4] approaches.

Traditionally, clustering is studied as an unsupervised task where no human supervision is provided to define the categories. More recently, it has been shown that clustering can be improved by considering various types of side information such as pair-wise constraints specifying a pair of instances must or must not belong to the same cluster [5] [6]. While a substantial body of literature exists on the topic clustering with instance-level constraints, there has been limited work on clustering with constraints on a more global scale. In particular, we are interested in clustering with constraints on the desired cluster sizes, which we refer to as cardinality constraints.

Learning with label proportions has gained a momentum in recent years, e.g., [7–10] since obtaining label proportions is cheaper or more feasible than labeling samples. For example, a healthcare report on the proportion of patients in each disease is available but individual patient disease information is unavailable due to privacy [11]. Similarly, in political election data, the percent of supporters for each candidate is available but not their individual votes [8]. In [7], black carbon level estimation for individual particles is difficult with current bioengineering techniques but estimating black carbon level of a mass amount of particles aggregated over hours is feasible. In birdsong data, there are a large number of unlabeled bird recordings and the only available information is expert knowledge about the proportion of each bird syllable [12] [13]. Size information can also be obtained before performing clustering in several domains such as geoinformetric or document clustering [14].

In this paper, we introduce a discriminative graphical approach for clustering with cardinality constraints, which allows users to explicitly specify the desired size or proportion of clusters. We further introduce a novel Renyi-type entropy regularization to encourage crisp clustering solutions. Treating the cluster labels as latent variables, we propose an expectation maximization algorithm for maximum likelihood estimation of the model parameters that uses exact inference for two-class problems and an efficient Gaussian approximation for multi-class clustering problems. Experiments on synthetic and real data demonstrate that our method is highly competitive in comparison to the current state-of-the-art, achieving superior or comparable performance.

2. RELATED WORKS

Our proposed method builds a discriminative model for clustering. There are several existing discriminative probabilistic frameworks for clustering. In [15], conditional entropy is maximized, however, this objective usually leads to small number of clusters. [16] avoids that by proposing a posterior regularization with class balance. Due to potentially large number of clusters, [17] proposes a regularization technique to smooth out the boundary. Compared to [17], instead of approximately minimizing the different of posterior distribution with the class balance constraint, we directly force cardinality of each clusters as an observation and exactly learn parameter satisfying the cardinality constraint.

SVM-based approaches have also been considered for discriminative clustering, e.g., maximum margin based clustering [18], where the task is finding a labeling on the data such that assuming this labeling, the margin learnt by SVM is maximized. There are several variations of MMC, such as [19], [20], and [21]. Different from them, we use a probabilistic approach with probability support.

Cluster sizes have been previously considered in the literature, which mostly focused on acquiring balanced cluster sizes or heuristically dealing with unbalanced cluster sizes. In [22], a general framework is proposed to make clustering algorithms produce balanced clusters. Qian and Saligrama [23] changed the way of constructing graph for spectral clustering by ranking nodes based on the density level around it to deal with small size clusters. Size constraints have also been considered by Zhu et al. where an approximate solution is acquired using linear integer programming [14]. In contrast, our work uses maximum likelihood estimation with exact inference for two classes case to enforce cluster size constraints.

3. PROBLEM FORMULATION AND MODEL

Problem formulation. Consider a set of unlabeled data instances denoted by $\mathbf{X}$. Specifically, let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, where

This work is partially supported by the National Science Foundation grants CCF-1254218, DBI-1356792, and IIS-1055113.

$\mathbf{x}_i \in \mathbb{X} \subseteq \mathbb{R}^d$ is the ith instance. The unknown label (cluster) of the ith instance $\mathbf{x}_i$ is given by y_i, for $i = 1, 2, \ldots, n$, where $y_i \in \mathbb{Y} = \{1, 2, \ldots, C\}$ and C is the number of clusters. Let N_c denote the number of samples in the c-th cluster, i.e., $N_c = \sum_{i=1}^{n} \mathbb{I}[y_i = c]$, where $\mathbb{I}[\cdot]$ denotes the indicator function, taking value 1 if its argument is true and 0 otherwise. As side information, we assume that the desired number of instances for each cluster $\mathbf{N} = [N_1, N_2, \ldots, N_C]^T$ are provided. Our goal is to learn a discriminative classifier that maps a sample in $\mathbb{X}$ to a label in $\mathbb{Y}$ given the feature vectors $\mathbf{X}$ and $\mathbf{N}$ as inputs.

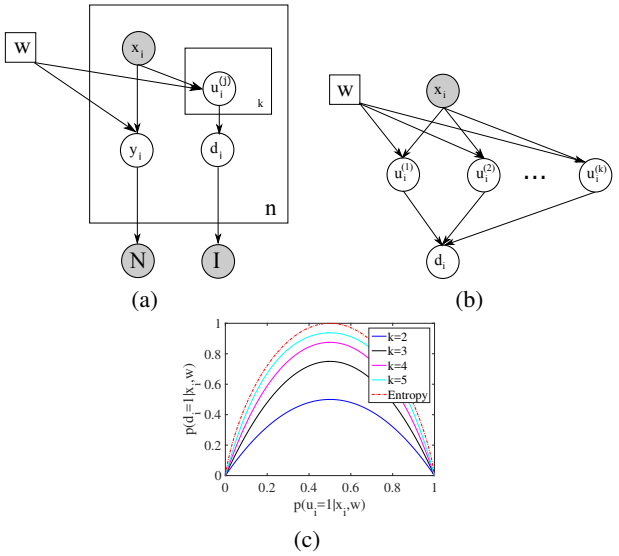

Fig. 1. (a) The proposed discriminative clustering probabilistic graphical model. Observed variables are shaded. (b) A graphical model for the disagreement indicator based on k independent randomly generate labels from the same sample. (c) The value of $p(d_i = 1|\mathbf{x}_i, \mathbf{w})$ vs. $p(u_i = 1|\mathbf{x}_i, \mathbf{w})$ for different k values as compared to the entropy function.

3.1. The proposed model

The proposed graphical model involves three key components, namely, the hidden labels, the observed cluster cardinalities, and the observed disagreement. We proceed with the description of the aforementioned components.

Hidden cluster labels. The probability of the label y_i of the ith instance $\mathbf{x}_i$ follows a multi-nomial logistic regression model:

$$p(y_i = c|\mathbf{x}_i, \mathbf{w}) = \frac{e^{\mathbf{w}_c^T \mathbf{x}_i}}{\sum_{c=1}^{C} e^{\mathbf{w}_c^T \mathbf{x}_i}}, \quad (1)$$

where $\mathbf{w}_c \in \mathbb{R}^d$, for $1 \leq c \leq C$, and $\mathbf{w} = [\mathbf{w}_1^T, \ldots, \mathbf{w}_C^T]^T$ is the classifier parameter vector. We assume that all instances $\mathbf{x}_i$ for $i = 1, 2, \ldots, n$ are i.i.d. given feature vectors and $\mathbf{w}$.

Cluster cardinalities. As stated in the problem formulation, we assume that number of instances in each cluster, i.e., $\mathbf{N} = [N_1, \ldots, N_C]$ where $N_c = \sum_{i=1}^{n} \mathbb{I}[y_i = c]$ are specified as part of the input. Consequently, the probability model for the observed cardinalities is deterministically given by

$$p(\mathbf{N}|\mathbf{y}) = \prod_{c=1}^{C} \mathbb{I}[N_c = \sum_{i=1}^{n} \mathbb{I}[y_i = c]\,], \quad (2)$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_n]$. In other words, the cardinality N_c of the cth cluster is equal to the total number of instances in the cluster, for all c.

Cluster crispness via disagreement. To provide a control on how well-separated and crisp the clusters should be, we introduce the notion of disagreement as follows. For each instance $\mathbf{x}_i$, we consider a disagreement indicator $d_i \in \{0, 1\}$ that can be viewed as an indirect measure of the confidence of the predictions that model (1) provides for instance $\mathbf{x}_i$. Specifically, given $\mathbf{x}_i$ and $\mathbf{w}$, k independent labels $u_i^{(1)}, u_i^{(2)}, \ldots, u_i^{(k)}$ are sampled according to the multinomial logistic regression model in (1), as shown in Fig. 1(b). Then, d_i is the indicator specifying that those k labels are not identical, i.e., $d_i = 1 - \mathbb{I}[u_i^{(1)} = u_i^{(2)} = \cdots = u_i^{(k)}]$. The intuition is that good clustering solutions will be well separated and hence the number of instances whose $d_i = 1$ should be small. The probability $p(d_i = 1|\mathbf{x}_i, \mathbf{w})$ can be obtained by marginalizing over $u_1, \ldots, u_k$ as follows

$$p(d_i = 1|\mathbf{x}_i, \mathbf{w}) = 1 - \sum_{c=1}^{C} p(u_i = c|\mathbf{x}_i, \mathbf{w})^k = 1 - \frac{\sum_{c=1}^{C} e^{k \times \mathbf{w}_c^T \mathbf{x}_i}}{(\sum_{c=1}^{C} e^{\mathbf{w}_c^T \mathbf{x}_i})^k}.$$

Consider the special case of $k = 2$ in a two-cluster setting. In this scenario, $p(d_i = 1|\mathbf{x}_i, \mathbf{w}) = 1 - p(u_i = 1|\mathbf{x}_i, \mathbf{w})^2 - (1 - p(u_i = 1|\mathbf{x}_i, \mathbf{w}))^2$. This probability is minimized when $p(u_i = 1|\mathbf{x}_i, \mathbf{w}) = 0$ or $p(u_i = 1|\mathbf{x}_i, \mathbf{w}) = 1$ (see Fig. 1(c)), i.e., the case in which a classifier provides deterministic predictions of the label for $\mathbf{x}_i$. In [24], a similar effect is achieved by entropy minimization, i.e., by minimizing $-\sum_{i=1}^{C} p_i \log p_i + (1 - p_i) \log(1 - p_i)$ where $p_i = p(u_i = 1|\mathbf{x}_i, \mathbf{w})$. A comparison between the entropy and the probability $p(d_i = 1|\mathbf{x}_i, \mathbf{w})$ as a function of $p(u_i = 1|\mathbf{x}_i, \mathbf{w})$ for different values of k is shown in Fig. 1(c). Aggregating the disagreements, we introduce a single binary variable I, which takes the value 1 when the total number of disagreements in the data is smaller than a given pre-specified value $m \in \{0, 1, 2, \ldots, n\}$ and zero otherwise:

$$p(I = 1|\mathbf{d}) = \mathbb{I}[\sum_{i=1}^{n} d_i \leq m],$$

where $\mathbf{d} = [d_1, d_2, \ldots, d_n]$. By requiring that $I = 1$ and varying the tuning parameter m, we can control the amount of label disagreement in the data, and hence the confidence and crispness of the learned clusters.

The cluster crispness is designed for finding well separated clusters whereas the cluster cardinalities is designed to avoid irregular cluster solution that also has high cluster crispness, e.g., there are only one big cluster and remaining clusters are empty. We propose to solve the clustering problem using maximum likelihood estimation on the aforementioned model to obtain $\mathbf{w}$, which is consistent with user-specified cluster cardinalities $\mathbf{N}$ and the disagreement count indicator $I = 1$.

4. INFERENCE

Maximum likelihood is used to infer model parameters. The log-likelihood of the model is $\log p(I, \mathbf{N}, \mathbf{X}|\mathbf{w}) = \log p(I, \mathbf{N}|\mathbf{X}, \mathbf{w}) + \log p(\mathbf{X}|\mathbf{w})$. Since the probability model for $\mathbf{X}$ is independent of $\mathbf{w}$, maximizing the log-likelihood is achieved by maximizing the conditional log-likelihood $\mathbf{L}(\mathbf{w}) = \log p(I, \mathbf{N}|\mathbf{X}, \mathbf{w})$. To address the challenges in the direct maximization of the log-likelihood, we consider the Expectation Maximization approach [25]. The EM auxiliary function is given by $Q(\mathbf{w}, \mathbf{w}') = E_{\mathbf{y},\mathbf{u}|I,\mathbf{N},\mathbf{w}'} \log p(I, \mathbf{N}, \mathbf{y}, \mathbf{u}|\mathbf{X}, \mathbf{w})$,

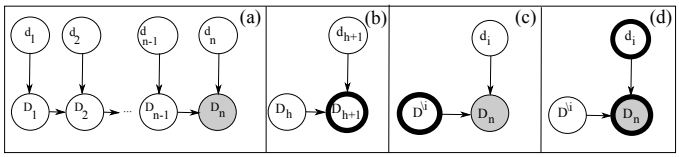

Fig. 2. Dynamic programming step for computing $p(u_i = c|I, \mathbf{X}, \mathbf{w}')$. Shaded nodes are observed. Bolded nodes are currently considered.

where $\mathbf{y}$ and $\mathbf{u} = [u_1^{(1)}, u_1^{(2)}, \ldots, u_1^{(k)}, \ldots, u_n^{(1)}, u_n^{(2)}, \ldots, u_n^{(k)}]$ are the hidden variables and I and $\mathbf{N}$ are the observations. Variable $\mathbf{d}$ is not considered as hidden since $\mathbf{d}$ can be deterministically computed from $\mathbf{u}$. To compute Q, we begin by deriving the complete log-likelihood

$$\begin{aligned}\mathbf{L}_c(\mathbf{w}) &= \log p(I, \mathbf{N}, \mathbf{y}, \mathbf{u}|\mathbf{X}, \mathbf{w}) \\ &= \log p(\mathbf{N}|\mathbf{y}) + \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) + \log p(I|\mathbf{u}) + \log p(\mathbf{u}|\mathbf{X}, \mathbf{w}).\end{aligned} \tag{3}$$

Consequently, the auxiliary function can be derived based on the complete log-likelihood as

$$\begin{aligned}Q(\mathbf{w}, \mathbf{w}') &= E_{\mathbf{y},\mathbf{u}|I,\mathbf{N},\mathbf{w}'}\Big[\mathbf{L}_c(\mathbf{w})\Big] \\ &= \zeta + \sum_{i=1}^{n}\Big[[\sum_{c=1}^{C} p(y_i = c|\mathbf{N}, \mathbf{X}, \mathbf{w}')\mathbf{w}_c^T\mathbf{x}_i - \log(\sum_{c=1}^{C} e^{\mathbf{w}_c^T\mathbf{x}_i})] \\ &\quad + k \times [\sum_{c=1}^{C} p(u_i = c|I, \mathbf{X}, \mathbf{w}')\mathbf{w}_c^T\mathbf{x}_i - \log(\sum_{c=1}^{C} e^{\mathbf{w}_c^T\mathbf{x}_i})]\Big],\end{aligned} \tag{4}$$

where ζ is a constant w.r.t. $\mathbf{w}$. We use generalized EM [25] in which the maximization of the auxiliary function $Q(\mathbf{w}, \mathbf{w}')$ w.r.t. $\mathbf{w}$ is replaced by an increase of $Q(\mathbf{w}, \mathbf{w}')$ w.r.t. $\mathbf{w}$. Consequently, the EM iteration is given by the following two steps:
E-step: Compute $p(y_i = c|\mathbf{N}, \mathbf{X}, \mathbf{w}^{(h)})$ and $p(u_i = c|I, \mathbf{X}, \mathbf{w}^{(h)})$, for $1 \leq i \leq n, 1 \leq c \leq C$.
M-step: Find $\mathbf{w}^{(h+1)}$ s.t. $Q(\mathbf{w}^{(h+1)}, \mathbf{w}^{(h)}) \geq Q(\mathbf{w}^{(h)}, \mathbf{w}^{(h)})$ where h is the current EM iteration.

4.1. E-step

In this section, we present methods to compute the probabilities in the E-step.

4.1.1. Computing $p(u_i = c|I, \boldsymbol{X}, \boldsymbol{w}')$ using dynamic programming

Let D_h and $D^{\setminus i}$ be the total number of disagreements in the first h samples and the total number of disagreements excluding the ith sample, respectively, such that $D_h = \sum_{j=1}^{h} d_j$ and $D^{\setminus i} = \sum_{j=1 \neq i}^{n} d_j = D_n - d_i$. Using the D_h's we form a chain model that allows for an efficient computation of the probability $p(u_i = c|I, \mathbf{X}, \mathbf{w}')$ using the following procedure. These steps are illustrated in Fig. 2.
Step 1. Initialize $p(D_1 = a|\mathbf{X}, \mathbf{w}') = p(d_1 = a|\mathbf{x}_1, \mathbf{w}')$ for $a \in \{0, 1\}$, and $p(d_1 = a|\mathbf{X}, \mathbf{w}') = 0$ for $a > 1$. Intuitively, the step initializes the probability of disagreement at the first instance.
Step 2. Dynamically compute $p(D_{h+1}|\mathbf{X}, \mathbf{w}')$ as in Fig. 2(b): $p(D_{h+1} = a|\mathbf{X}, \mathbf{w}') = p(d_{h+1} = 0|\mathbf{x}_{h+1}, \mathbf{w}')p(D_h = a|\mathbf{X}, \mathbf{w}') + p(d_{h+1} = 1|\mathbf{x}_{h+1}, \mathbf{w}')p(D_h = a - 1|\mathbf{X}, \mathbf{w}')$, for $0 \leq a \leq m$ and $0 \leq h \leq n - 1$. Intuitively, if there are a disagreements among the first $h + 1$ instances, then there are two possibilities that there are a or $a - 1$ disagreements among the first h instances since the disagreement at the (h+1)th instance is a binary value.
Step 3. Compute $p(D^{\setminus i}|\mathbf{X}, \mathbf{w}')$ using the forward and substitution method, as in Fig. 2(c), as follows

$$\begin{aligned}p(D^{\setminus i} = 0|\mathbf{X}, \mathbf{w}') &= \frac{p(D = 0|\mathbf{X}, \mathbf{w}')}{p(d_i = 0|\mathbf{x}_k, \mathbf{w}')}, \\ p(D^{\setminus i} = a + 1|\mathbf{X}, \mathbf{w}') &= \frac{p(D = a + 1|\mathbf{X}, \mathbf{w}')}{p(d_i = 0|\mathbf{x}_i, \mathbf{w}')} \\ &\quad - \frac{p(d_i = 1|\mathbf{x}_i, \mathbf{w}')p(D^{\setminus i} = a|\mathbf{X}, \mathbf{w}')}{p(d_i = 0|\mathbf{x}_i, \mathbf{w}')},\end{aligned} \tag{5}$$

for $0 \leq a \leq m - 1$ and $0 \leq i \leq n$. Intuitively, the distribution of the number of disagreements among all instances except the ith instance can be computed from that distribution among all instances.
Step 4. Compute $p(D^{\setminus i} \leq m|\mathbf{X}, \mathbf{w}') = \sum_{0 \leq a \leq m} p(D^{\setminus i} = a|\mathbf{X}, \mathbf{w}')$.
Step 5. Finally, compute $p(u_i = c|I = 1, \mathbf{X}, \mathbf{w}')$, as in Fig. 2(d), using conditional rule

$$\begin{aligned}&p(u_i = c|I = 1, \mathbf{x}_i, \mathbf{w}') = p(u_i = c|D \leq m, \mathbf{x}_i, \mathbf{w}') \\ &= \frac{p(u_i = c, D \leq m|\mathbf{X}, \mathbf{w}')}{\sum_{c=1}^{C} p(u_i = c, D \leq m|\mathbf{X}, \mathbf{w}')},\end{aligned} \tag{6}$$

where $p(u_i = c, D \leq m|\mathbf{X}, \mathbf{w}')$=$p(u_i = c|\mathbf{x}_k, \mathbf{w}')\Big[p(u_i = c|\mathbf{x}_k, \mathbf{w}')^{k-1}p(D^{\setminus i} \leq m|\mathbf{X}, \mathbf{w}')$+$(1-p(u_i = c|\mathbf{x}_k, \mathbf{w}'))^{k-1}p(D^{\setminus i} \leq m - 1|\mathbf{X}, \mathbf{w}')\Big]$. The computational complexity for computing $p(u_i = c|I, \mathbf{X}, \mathbf{w}')$ is $O(mn)$. This can be obtained from the fact that state probabilities are computed for n instances and each state takes $m + 1$ values.

4.1.2. Computing $p(y_i = c|\boldsymbol{N}, \boldsymbol{X}, \boldsymbol{w}')$ using Gaussian approximation

In this section, first, we present the dynamic programming method to compute $p(y_i = c|\mathbf{N}, \mathbf{X}, \mathbf{w}')$ and its computational challenge. Then, we show how to overcome that challenge using Gaussian approximation.

Denote $\mathbf{N}^h = [N_1^h, N_2^h, \ldots, N_C^h]$, $\mathbf{N}^{\setminus i} = [N_1^{\setminus i}, N_2^{\setminus i}, \ldots, N_C^{\setminus i}]$ as the number of samples in each cluster from the first to the kth sample and the number of samples in each cluster excluding the ith sample, i.e., $N_c^{\setminus i} = \sum_{j=1 \neq i}^{n} \mathbb{I}[y_j = c]$, respectively. The probability $p(y_i = c|\mathbf{N}, \mathbf{X}, \mathbf{w}')$ is computed using the conditional probability definition $p(y_i = c|\mathbf{N}, \mathbf{X}, \mathbf{w}') = \frac{p(y_i=c,\mathbf{N}|\mathbf{X},\mathbf{w}')}{\sum_{l=1}^{C} p(y_i=l,\mathbf{N}|\mathbf{X},\mathbf{w}')}$, where $p(y_i = c, \mathbf{N}|\mathbf{X}, \mathbf{w}')$ is computed by marginalizing $p(y_i = c, \mathbf{N} = \mathbf{v}|\mathbf{X}, \mathbf{w}') = p(y_i = c|\mathbf{x}_i, \mathbf{w}')p(\mathbf{N}^{\setminus i} = \mathbf{v} - \mathbf{e}_c|\mathbf{X}, \mathbf{w}')]$ where $\mathbf{e}_c \in \{0, 1\}^C$ is the canonical vector such that $\mathbf{e}_c(j) = 1$ for $j = c$ and zero otherwise.

A dynamic programming approach to compute $p(\boldsymbol{N}^{\setminus i}|\boldsymbol{X}, \boldsymbol{w}')$. The probability $p(\mathbf{N}^{\setminus i} = \mathbf{v} - \mathbf{e}_c|\mathbf{X}, \mathbf{w}')$ is computed using the dynamic programming process as follows.
Step 1. Initialize $p(\mathbf{N}^1 = \mathbf{v}|\mathbf{X}, \mathbf{w}')$ from $p(y_1|\mathbf{x}_1, \mathbf{w}')$ as follows

$$p(\mathbf{N}^1 = \mathbf{v}|\mathbf{X}, \mathbf{w}') = \begin{cases} p(y_1 = c|\mathbf{x}_1, \mathbf{w}') & \text{if } \mathbf{v} = \mathbf{e}_c. \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

Step 2. Dynamically compute $p(\mathbf{N}|\mathbf{X}, \mathbf{w}')$ using $p(\mathbf{N}^{(h+1)} = \mathbf{v}|\mathbf{X}, \mathbf{w}') = \sum_{c=1}^{C} p(y_{h+1} = c|\mathbf{x}_{h+1}, \mathbf{w}')p(\mathbf{N}^h = \mathbf{v} - \mathbf{e}_c|\mathbf{X}, \mathbf{w}')$. Note that $\mathbf{v} \in \mathbb{Z}^C$ and $0 \leq \mathbf{v}_j \leq h + 1, \forall 1 \leq j \leq C$.

Step 3. Compute $p(\mathbf{N}^{\setminus i}|\mathbf{X}, \mathbf{w}')$ using the forward and substitution method in the C-dimensional space similar to (5).

The computational complexity for computing Step 2 is $O(n^C)$. When the number of clusters C is large, the aforementioned exact approach for computing $p(y_i = c|\mathbf{N}, \mathbf{X}, \mathbf{w}')$ is computationally prohibitive. To address this problem, we present an alternative approach using the following Gaussian approximation computation.

A Gaussian approximation approach to compute $p(\mathbf{N}^{\setminus i}|\mathbf{X}, \mathbf{w}')$. Following the central limit theorem, since $\mathbf{N}^{\setminus i} = \sum_{j=1\neq i}^{n} \mathbf{v}_j$ with $\mathbf{v}_j = [\mathbb{I}[y_j = 1], \mathbb{I}[y_j = 2], \ldots, \mathbb{I}[y_j = C]]^T$ and $\mathbf{v}_j$'s are independent with finite variance, $\mathbf{N}^{\setminus i}$ is asymptotically Gaussian for sufficiently large n [26]. For each jth instance, define $\mu_j \in \mathbb{R}^C$ and $\Sigma_j \in \mathbb{R}^{C\times C}$ as $\mu_j(r) = p(y_j = r|\mathbf{x}_j, \mathbf{w}')$ and $\Sigma_j(r, s) = \mathbb{I}[r = s]p(y_j = r|\mathbf{x}_j, \mathbf{w}') - p(y_j = r|\mathbf{x}_j, \mathbf{w}')p(y_j = s|\mathbf{x}_j, \mathbf{w}')$, where $1 \leq r, s \leq C$. We approximate $\mathbf{N}^{\setminus i}$ as Gaussian: $\mathcal{N}(\mu^{\setminus i}, \Sigma^{\setminus i})$ where $\mu^{\setminus i} = \sum_{j=1\neq i} \mu_j$ and $\Sigma^{\setminus i} = \sum_{j=1\neq i} \Sigma_j$ and therefore $p(\mathbf{N}^{\setminus i} = \mathbb{N}|\mathbf{X}, \mathbf{w}')$ $=2\pi^{-\frac{C}{2}} |\Sigma^{\setminus i}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\mathbb{N}-\mu^{\setminus i})^T(\Sigma^{\setminus i})^{-1}(\mathbb{N}-\mu^{\setminus i})}$. The computational complexity using Gaussian approximation is $O(nC)$.

4.2. M-step

We use gradient ascent with backtracking line-search [27] in M-step. Specifically, $\mathbf{w}^{(h+1)} = \mathbf{w}^{(h)} + \eta \frac{\partial Q(\mathbf{w}, \mathbf{w}^{(h)})}{\partial \mathbf{w}}\Big|_{\mathbf{w}=\mathbf{w}^{(h)}}$, where the gradient is computed as $\sum_{i=1}^{n} \mathbf{x}_i[p(y_i = c|\mathbf{N}, \mathbf{X}, \mathbf{w}^{(h)}) - p(y_i = c|\mathbf{x}_i, \mathbf{w})]$+ $\sum_{i=1}^{n} k \times \mathbf{x}_i[p(u_i = c|I, \mathbf{X}, \mathbf{w}^{(h)}) - p(y_i = c|\mathbf{x}_i, \mathbf{w})]$.

5. EXPERIMENTS

In this section, we evaluate the performance of the proposed algorithm DCCC, and compare with several clustering algorithms including K-mean, Maximum Margin Clustering (MMC), and discriminative clustering with Regularization Information Maximization (RIM). As mentioned in Section 1, we assume the size constraints N is known for every dataset. Note that both the proposed DCCC method and RIM benefit the size constraints as input. We use normalized mutual information NMI as the evaluation metric [28] [29] [30].

5.1. Experiments on MNIST handwritten dataset

Datasets and setting. We generate five datasets, each containing two classes of digits: 12, ..., 78, and 90. Each dataset contains 200 digits generated uniformly from two classes. Because the digits in the MNIST datasets are reasonably well separated, for this set of experiments we set the disagreement parameter $m = 0$. We use L_2 regularization for the proposed algorithm with parameter λ selected in $\{10, 20, 50, 100, \ldots, 10^3\}$. We compare DCCC with RIM, K-means, and MMC. Specifically, the parameter λ for RIM is searched over the set $\{10, 20, 50, 100, \ldots, 10^3\}$. The parameter ℓ for MMC is searched over the set $\{0.01, 0.02, 0.04, 0.1, 0.2, 0.4\}$ and C is set to 0.001. For each parameter setting, each algorithm is initialized 10 times and the model that yields the optimal value of the objective is selected and its performance reported. Finally, the parameters are selected post-hoc to maximize the performance. We perform experiments on both implementations of DCCC: using exact dynamic programming computation with $O(n^2)$ time complexity and using Gaussian approximation with $O(n)$ time complexity.

Results and analysis. The NMI results are presented in Table 1. In comparison to RIM, we observe that DCCC achieves comparable or superior performance for most of the datasets. This may be due to the fact that exact inference is used in our algorithm to increase the likelihood at each step, whereas RIM uses approximations in maximizing the likelihood and regularization with the reference distribution. The performance of the implementation for DCCC using Gaussian approximation DCCC-Gaussian (DCCC-G, for short) is similar to that of the exact implementation using dynamic programming DCCC-Dynamic (DCCC-D, for short). This may be due to the fact that there are large number of samples making the approximate distribution close to the exact distribution, based on the central limit theorem [26]. Compared to DCCC and RIM, the results of K-means and MMC are often considerably lower.

Datasets	1 2	3 4	5 6	7 8	9 0
DCCC-D	0.70	**0.93**	**0.72**	**0.89**	**0.93**
DCCC-G	0.70	**0.93**	**0.72**	**0.89**	**0.93**
RIM	**0.73**	0.89	0.69	0.88	**0.93**
MMC	0.64	0.81	0.71	0.76	0.90
Kmeans	0.46	0.81	0.56	0.79	0.81

Table 1. NMI results of DCCC, RIM, MMC, and K-means on MNIST datasets.

Datasets	HJA bird song	MSCV2	Voc12
DCCC-G	**0.40**	**0.31**	**0.12**
RIM	0.39	0.25	0.11
K-means	0.06	0.13	0.02

Table 2. NMI results of DCCC, RIM, and K-means on real datasets.

5.2. Experiments on bird song and image annotation datasets

Datasets and setting. We consider three real-world multi-class datasets including HJA bird song (13-class), MSCV2 image annotation (19-class), and Voc12 (20-class) [13] [31]. For these datasets, we uniformly select 400 samples for training and use the remaining samples as a validation set for parameter tuning. In particular, we use the log-likelihood on the validation set to select the parameter for our model and use the objective value on the validation set for tuning the parameter for RIM. Since the number of classes C is large, Gaussian implementation is used for DCCC. For the proposed method, m, λ, and k are selected from the set $\{10, ..., 50\}$, $\{0.1, 1, 10, 100, 1000\}$, and $\{2, 3\}$, respectively. For the regularization parameter of the RIM parameter, we use the same selection range as used for the MNIST data. We use the true class proportion to specify the cluster sizes for our method and the reference distribution for RIM. Since MMC is designed for two classes problem, we skip this algorithm.

Results and analysis. The NMI results are presented in Tables 2. We observe that the performance of DCCC is significantly higher than that of RIM on MSCV2 dataset, and a little higher on HJA and Voc12 datasets.

6. CONCLUSION

In this paper, we proposed a discriminative graphical model for clustering with cluster size constraints. The framework achieves cluster crispness and the desired class proportion using the maximum likelihood approach via EM. To overcome the computational challenge in the E-step, we introduce a novel Gaussian approximation. Experiments on real datasets demonstrated that our method is highly competitive compared to the current state-of-the-art methods.

7. REFERENCES

[1] C. M. Bishop, *Pattern recognition and machine learning*, springer, 2006.

[2] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[3] M. Ester, H. Kriegel, J. Sander, X. Xu, A. Idrissov, M. Nascimento, R. Ng, A. Thiagarajan, J. Biagioni, T. Gerlich, et al., "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, vol. 2, pp. 49–60.

[4] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Advances in neural information processing systems*, 2005, pp. 1537–1544.

[5] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al., "Constrained k-means clustering with background knowledge," in *Proceedings of the International Conference on Machine Learning*, 2001, vol. 1, pp. 577–584.

[6] Z. Li, J. Liu, and X. Tang, "Pairwise constraint propagation by semidefinite programming for semi-supervised classification," in *Proceedings of the International Conference on Machine Learning*, 2008, pp. 576–583.

[7] D. R. Musicant, J. M. Christensen, and J. F. Olson, "Supervised learning by training on aggregate outputs," in *International Conference on Data Mining*, 2007, pp. 252–261.

[8] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le, "Estimating labels from label proportions," *Journal of Machine Learning Research*, vol. 10, no. Oct, pp. 2349–2374, 2009.

[9] S. Rueping, "SVM classifier estimation from group probabilities," in *Proceedings of the International Conference on Machine Learning*, 2010, pp. 911–918.

[10] F. X. Yu, D. Liu, S. Kumar, T. Jebara, and S. F. Chang, "SVM for learning with label proportions," in *Proceedings of the International Conference on Machine Learning*, 2013.

[11] F. X Yu, K. Choromanski, S. Kumar, T. Jebara, and S.-F. Chang, "On learning from label proportions," *arXiv preprint arXiv:1402.5902*, 2014.

[12] L. Neal, F. Briggs, R. Raich, and X. Z. Fern, "Time-frequency segmentation of bird song in noisy acoustic environments," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 2012–2015.

[13] F. Briggs, X. Z. Fern, and R. Raich, "Rank-loss support instance machines for MIML instance annotation," in *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 534–542.

[14] S. Zhu, D. Wang, and T. Li, "Data clustering with size constraints," *Knowledge-Based Systems*, vol. 23, no. 8, pp. 883–889, 2010.

[15] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in neural information processing systems*, 2004, pp. 529–536.

[16] J. S. Bridle, A. J.R. Heading, and D. J.C. MacKay, "Unsupervised classifiers, mutual information and'phantom targets," in *Advances in neural information processing systems*, 1992, pp. 1096–1101.

[17] A. Krause, P. Perona, and R. G. Gomes, "Discriminative clustering by regularized information maximization," in *Advances in neural information processing systems*, 2010, pp. 775–783.

[18] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum margin clustering," in *Advances in neural information processing systems*, 2004, pp. 1537–1544.

[19] K. Zhang, I. W. Tsang, and J. T. Kwok, "Maximum margin clustering made practical," in *Proceedings of the International Conference on Machine Learning*, 2007, pp. 1119–1126.

[20] B. Zhao, F. Wang, and C. Zhang, "Efficient multiclass maximum margin clustering," in *Proceedings of the International Conference on Machine Learning*, 2008, pp. 1248–1255.

[21] Y.-F. Li, I. W. Tsang, J. T. Kwok, and Z.-Hua. Zhou, "Tighter and convex maximum margin clustering," in *International Conference on Artificial Intelligence and Statistics*, 2009, pp. 344–351.

[22] A. Banerjee and J. Ghosh, "Scalable clustering algorithms with balancing constraints," *Data Mining and Knowledge Discovery*, vol. 13, no. 3, pp. 365–395, 2006.

[23] J. Qian and V. Saligrama, "Spectral clustering with imbalanced data," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 3057–3061.

[24] R. Jenssen, K. E. Hild, D. Erdogmus, J. C. Principe, and T. Eltoft, "Clustering using renyi's entropy," in *Proceedings of the International Joint Conference on Neural Networks*, 2003, vol. 1, pp. 523–528.

[25] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, pp. 1–38, 1977.

[26] W. Feller, *An introduction to probability theory and its applications*, vol. 2, John Wiley & Sons, 2008.

[27] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.

[28] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *Neural Networks, IEEE Transactions on*, vol. 20, no. 2, pp. 189–201, 2009.

[29] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[30] M. Wu and B. Schölkopf, "A local learning approach for clustering," in *Advances in neural information processing systems*, 2006, pp. 1529–1536.

[31] M. Everingham, Luc Van G., C. K.I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.