Crowdclustering with Partition Labels

Junxiang Chen[1] Yale Chang[1] Peter Castaldi[2] Michael Cho[2] Brian Hobbs[2] Jennifer Dy[1]
[1] Electrical and Computer Engineering Department, Northeastern University, Boston, MA, USA
[2] Channing Division of Network Medicine, Brigham and Women's Hospital, Boston, MA, USA

Abstract

Crowdclustering is a practical way to incorporate domain knowledge into clustering, by combining opinions from multiple domain experts. Existing crowdclustering methods analyze binary pairwise similarity labels. However, in some applications, experts might provide partition labels. If we convert partition labels into pairwise similarity, then it would be difficult to understand the relationships between clustering solutions from different experts. In this paper, we propose a crowdclustering model that directly analyzes partition labels. The proposed model adopts a novel approach based on a modified multinomial logistic regression model, which simultaneously learns the number of clusters and determines hyper-planes that partition samples into clusters. The proposed model also learns a mapping between the latent clusters and expert labels, revealing the agreements and disagreements between experts. Experiments on benchmark data demonstrate that the proposed model simultaneously learns the number of clusters and discovers the clustering structure. An experiment on disease subtyping problem illustrates that the proposed model helps us understand the agreement and disagreement between experts.

1 Introduction

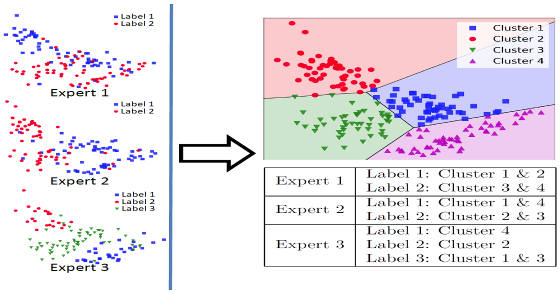
Clustering is a task of grouping objects into several categories, such that objects in the same category are similar; while objects from different categories are dissimilar. Because high-dimensional data is usually richly structured, there might exist multiple meaningful clustering solutions for the same data set[1, 2]. However, domain experts are oftentimes only interested in one particular solution for a specific purpose. Therefore, allowing domain experts to provide knowledge under semi-supervised settings is helpful in clustering tasks [3–5].

Due to the exploratory nature of clustering tasks, domain experts might not completely agree with each other. Therefore, one practical way to incorporate expert knowledge is to collect opinions from multiple experts and combine these opinions into a consensus clustering solution. Methods that find such consensus is known as crowdclustering [6–10].

Although crowdclustering methods have succeeded in practice; as far as we know, all existing methods analyze binary pairwise similarity labels, indicating whether a pair of samples are similar and should belong to the same cluster or not. But in some applications, domain experts directly give partition labels. Although it is straightforward to convert partition labels into pairwise similarity labels, using pairwise similarity labels might be less favoured; because in exploratory discovery tasks, we are usually not only interested in finding consensus that summarizes all expert opinions, we also want to understand the agreement and disagreement between experts, and how the consensus clusters learned are related to the observed expert labels. If we convert partition labels into pairwise similarity, it is more difficult to recover this information.

In this paper, we propose a probabilistic clustering model to overcome the limitations of existing methods. Unlike existing crowdclustering models, the proposed model directly makes use of partition labels rather than pairwise similarity. In the proposed model, we developed a novel approach based on a modified multinomial logistic regression model to generate the latent cluster memberships given observed features. This approach simultaneously determines the number of clusters, based on a “rich get richer” fashion; and partitions samples into clusters with hyper-planes. The proposed model also learns the mapping between the latent clusters

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.



Expert 1	Label 1: Cluster 1 & 2 Label 2: Cluster 3 & 4
Expert 2	Label 1: Cluster 1 & 4 Label 2: Cluster 2 & 3
Expert 3	Label 1: Cluster 4 Label 2: Cluster 2 Label 3: Cluster 1 & 3

Figure 1: The problem setup.

and expert labels, with a variant of Bayesian Classifier Combination (BCC) model, revealing the agreement and disagreement between experts.

We illustrate the problem setup in Figure 1. Given the features for samples and partition labels from 3 experts, the proposed model automatically discovers that there are 4 latent clusters and finds the hyper-planes that define these clusters. It also learns the relationships between the discovered clusters and each expert label, as described in the table.

1.1 Related Work

Several crowdclustering methods have been proposed in recent years. [6] and [7] analyze pairwise similarity labels given by the experts without accessing the features of the samples. [8–10] make use of both the pairwise similarity labels and the sample features to generate a clustering solution. Unlike these methods that analyze the pairwise similarity labels, the proposed method directly analyzes the partition labels. Since the proposed method learns a mapping between the clusters and the expert labels, it reveals the agreement and disagreement between experts.

Clustering ensemble [11–18] is also closely related to this topic, since it involves generating a consensus clustering result from several clustering solutions. Unlike crowdclustering methods, where clustering solutions are provided by domain experts, clustering ensemble methods analyze the clustering solutions generated by basic clustering models such as k-means. Although clustering ensemble techniques can also be used to generate consensus results based on the expert labels; these methods have difficulties in out-of-sample prediction, because the expert labels are oftentimes not available for test samples. The proposed method learns a linear discriminative model and can be used to predict test samples based on the observed features.

1.2 Contributions

The contributions of this paper are summarized as follows:

1. The proposed model makes use of partition labels rather than pairwise similarity labels. This allows us to better understand the relations between labels from different experts.
2. We developed a novel approach to generate latent clusters given the observed features. This approach simultaneously determines the number of clusters and partitions samples into clusters.
3. We test the proposed method on both benchmark data sets and a real-world data set to demonstrate its effectiveness and usefulness.

2 Method

In this section we introduce the proposed model. We first introduce some notations used in this paper in Section 2.1. Then, we introduce the modified multinomial logistic regression model in Section 2.2. This model simultaneously determines the number of clusters, based on a "rich get richer" fashion; and partitions samples into clusters with hyper-planes. We explain the detail about the "rich get richer" property in Section 2.3. In Section 2.4, we introduce how we generate the expert labels, based on a variant of Bayesian Classifier Combination (BCC) model. The overall model is summarized in Section 2.5.

2.1 Notations

We start by introducing some notations. We assume that the data set contains N samples with D features. We let $\boldsymbol{\phi}_n \in \mathbb{R}^{D+1}$ be a $(D+1)$-dimensional vector associated with each sample $n \in \{1 \dots N\}$, where the first D dimensions in this vector are the observed features for the n-th sample, and the $(D+1)$-th dimension is a constant 1. We will train a linear model with $\boldsymbol{\phi}_n$ and the corresponding weight parameter for the $(D+1)$th dimension plays a role of the bias term. We denote features for all samples as $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_n\}_{n=1}^N$.

We assume that labels are provided by M experts. We let $y_n^{(m)} \in \{1 \dots J_m\}$ with $m \in \{1 \dots M\}$ be the labels given by the m-th expert for the n-th sample, where J_m represents the number of clusters the m-th expert chooses to partition the data set. Note that J_m might differ across $m \in \{1 \dots M\}$, since experts might choose to partition the data set into different number of clusters. We let $\mathbf{Y} = \{\{y_n^{(m)}\}_{n=1}^N\}_{m=1}^M$ represent labels from all experts.

2.2 Modified Multinomial Logistic Regression

Now we introduce how we generate the latent cluster indicators, such that we simultaneously determine the number of clusters and partition samples into clusters.

The number of clusters is usually a predefined parameter for clustering algorithms, including several crowdclustering methods [7–10]. This number might not be easy to determine for crowdclustering, especially when experts partition data into different number of clusters. One possible way to automatically determine this number is to apply Dirichlet Process (DP) [19, 20] as a prior for the cluster indicators in a *generative* model. However, a generative model requires strong assumptions about the distribution of the observed samples, but these assumptions are usually inaccurate in practice. Therefore, we decide to develop a *discriminative* model.

It is not straightforward to incorporate DP in a discriminative model. Therefore, we develop a novel approach, based on a modified multinomial logistic regression model, to automatically learn the number of clusters. This approach is inspired from the "rich get richer" fashion adopted by DP, such that big clusters that already have many members are more likely to be assigned more new members. To achieve this, we define $z_n \in \{1 \dots K\}$ to be the cluster indicator for the n-th sample, where K is a predefined integer parameter that represents the maximum possible number of clusters. We let K have a large value such that $K = 50$ in the experiment, i.e., samples can potentially be partitioned into up to 50 clusters. Because of the "rich get richer" property as we describe in details in Section 2.3, only a few clusters will remain non-empty after we train the model.

We let z_n follow a categorical distribution such that

$$z_n | \mathbf{W}, \mathbf{\Phi} \sim \text{Categorical}(\boldsymbol{\pi}_n). \tag{1}$$

In this equation, $\boldsymbol{\pi}_n$ is a K dimensional non-negative vector, such that $\sum_{k=1}^{K} \pi_{nk} = 1$. π_{nk} gives the probability that the n-th sample belongs to cluster k, which is defined as

$$\pi_{nk} = \frac{\exp(\mathbf{w}_k^T \boldsymbol{\phi}_n + \lambda \mathbf{w}_k^T \mathbf{w}_k)}{\sum_{i=1}^{K} \exp(\mathbf{w}_i^T \boldsymbol{\phi}_n + \lambda \mathbf{w}_i^T \mathbf{w}_i)} \tag{2}$$

where $\mathbf{w}_k \in \mathbb{R}^{D+1}$ with $k \in \{1 \dots K\}$ is a $(D+1)$-dimensional vector, each element of which represents the weight for each feature in $\boldsymbol{\phi}_n$ and λ is a predefined non-negative parameter. In equation (1), we use $\mathbf{W}$ to denote all the weight vectors $\{\mathbf{w}_k\}_{k=1}^{K}$.

We assign a Gaussian prior for each $\mathbf{w}_k$ such that

$$\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}), \tag{3}$$

where σ^2 is the variance parameter.

Note that if we let $\lambda = 0$, then the model becomes the regular logistic regression. We modified logistic regression by introducing an additional $\lambda \mathbf{w}_k^T \mathbf{w}_k$ term with $\lambda > 0$, such that the model exhibits the "rich-get-richer" property as described in Section 2.3.

2.3 The "Rich-Get-Richer" Property

We determine the number of clusters via the "rich-get-richer" property, i.e., big clusters that already have many members are more likely to be assigned more new members; because we modified the loss function of the logistic regression model as shown in Equation (2).

Equation (2) indicates that given the same prediction performance of the linear model (determined by $\mathbf{w}_k^T \boldsymbol{\phi}_n$), a sample is more likely to be assigned to a cluster with larger $\mathbf{w}_k^T \mathbf{w}_k$. Note that $\lambda \mathbf{w}_k^T \mathbf{w}_k$ is not a function of the features $\boldsymbol{\phi}_n$; and serves as an additional non-negative bias term.

Now if we apply the Expectation Maximization (EM) [21] to learn the maximum a posteriori probability (MAP) estimator for $\mathbf{W}$, then the derived maximization step that updates $\mathbf{W}$ is given as

$$\begin{aligned} \widehat{\mathbf{W}} = \arg\max_{\mathbf{W}} - \sum_{k=1}^{K} \left(\frac{1}{2\sigma^2} - \lambda \sum_{n=1}^{N} \mathbb{E}[\mathbf{1}(z_n = k)] \right) \mathbf{w}_k^T \mathbf{w}_k \\ + \sum_{k=1}^{K} \sum_{n=1}^{N} \mathbb{E}[\mathbf{1}(z_n = k)] \mathbf{w}_k^T \boldsymbol{\phi}_n - \sum_{n=1}^{N} \log \sum_{i=1}^{K} \exp \left(\mathbf{w}_i^T \boldsymbol{\phi}_n + \lambda \mathbf{w}_i^T \mathbf{w}_i \right). \end{aligned} \tag{4}$$

where $\mathbb{E}$ represents an expected value is taken with respect to the posterior distribution of z_n. $\mathbf{1}(z_n = k)$ is an indicator function that returns 1, only if $z_n = k$; and returns 0, otherwise. Note that $\sum_{n=1}^{N} \mathbb{E}[\mathbf{1}(z_n = k)]$ represents the expected number of samples assigned to cluster k. By observing the first term in Equation (4), we conclude that for the cluster k that contains more members, the model penalizes less with respect to $\mathbf{w}_k^T \mathbf{w}_k$. Therefore, if a cluster k has more members, it tends to have a larger $\mathbf{w}_k^T \mathbf{w}_k$ value.

From Equation (4), we see that big clusters with more members tend to have larger $\mathbf{w}_k^T \mathbf{w}_k$. As shown in Equation (2), samples are more likely to be assigned to clusters with larger $\mathbf{w}_k^T \mathbf{w}_k$. Therefore, in the EM iterations, large clusters that already have many members are more likely to be assigned more new members. This exhibits the "rich get richer" property. This property allows us to initialize the model with a large number of clusters (50 in our experiments). Similar to the updates in variational inference of Dirichlet process [20], only a few clusters remain non-empty when the optimization converges. The number of these non-empty clusters is the number of clusters that is automatically determined by the model.

In Equation (2), λ is a parameter that controls the trade off between prediction accuracy and cluster size. Larger λ makes Equation (2) depend more on the cluster size (represented by $\mathbf{w}_k^T \mathbf{w}_k$), but less on prediction accuracy (represented by $\mathbf{w}_k^T \boldsymbol{\phi}_n$) . In the experiments, we choose $\lambda = 1/(2N\sigma^2)$ to ensure that the coefficients for $\mathbf{w}_k^T \mathbf{w}_k$ in Equation (4) are non-positive and introduce $l2$ regularization for the optimization.

2.4 Generating Expert Labels

Now, we introduce how we model the expert labels $\mathbf{Y}$ given $\mathbf{Z}$, by learning a mapping from $\mathbf{Z}$ to $\mathbf{Y}$, with a variant of Bayesian Classifier Combination (BCC) model [17, 18].

Unlike the traditional BCC model that learns confusion matrices that describe the mapping between the clusters and expert labels, we model such relationship by assigning the cluster to the expert labels. This allows us to better understand the relationship between the cluster and the expert labels, as illustrated in the table shown in Figure 1. We let $t_k^{(m)} \in \{1 \dots J_m\}$ indicate which label the cluster k is assigned to for expert m, such that $t_k^{(m)} = j$ implies that cluster k is assigned to the j-th label given by expert m. We assign a uniform prior for $t_k^{(m)}$, such that

$$t_k^{(m)} \sim \text{Categorical}\left(\left[\frac{1}{J_m}, \dots, \frac{1}{J_m}\right]\right), \tag{5}$$

Note that for each expert m, each cluster k is assigned to exactly one label j, but this label j might be associated with multiple clusters, as illustrated in the table of Figure 1. This differs from traditional BCC model that describes such relationships with confusion matrices, and it would be difficult to interpret the relationships.

After assigning the cluster to the labels, we want to make sure that expert labels can be accurately predicted with the observed cluster indicator z_n. For example, if cluster k is assigned to label j for expert m and sample n belongs to cluster k, such that $t_k^{(m)} = j$ and $z_n = k$, then it should be very likely that expert m gives the label j for the n-th sample, i.e., the probability that $y_n^{(m)} = j$ is high. Therefore, if we define a non-negative J_m-dimensional vector $\boldsymbol{\eta}^{(mj)}$ with each element $\sum_{l=1}^{J_m} \eta_l^{(mj)} = 1$, to represent the conditional probability

$$\eta_l^{(mj)} \stackrel{\text{def}}{=} p\left(y_n^{(m)} = l | z_n = k \text{ and } t_k^{(m)} = j\right), \tag{6}$$

then it must be true that

$$\eta_j^{(mj)} \gg \eta_l^{(mj)}, \text{ for all } l \neq j. \tag{7}$$

Note that Equation (6) is equivalent to letting $y_n^{(m)}$ follow a mixture of categorical distribution such that

$$y_n^{(m)} | z_n, \mathbf{T} \sim \prod_{k=1}^{K} \prod_{j=1}^{J_m} \left\{\text{Categorical}\left(\boldsymbol{\eta}^{(mj)}\right)\right\}^{\mathbf{1}(t_k^{(m)}=j)\mathbf{1}(z_n=k)}. \tag{8}$$

We enforce Condition (7) by assigning a Dirichlet distribution prior for $\boldsymbol{\eta}^{(mj)}$, such that

$$\boldsymbol{\eta}^{(mj)} \sim \text{Dirichlet}\left(\boldsymbol{\Psi}^{(m)}\right), \tag{9}$$

where $\boldsymbol{\Psi}^{(m)} = \{\Psi_l^{(m)}\}_{l=1}^{J_m}$ is a J_m-elemental vector, each of whose elements defined as

$$\Psi_l^{(m)} = \begin{cases} \alpha, & \text{if } l = j \\ \beta, & \text{if } l \neq j. \end{cases} \tag{10}$$

α and β are concentration parameters for the Dirichlet distribution. In order to make Condition (7) be satisfied, we chose $\alpha \gg \beta$.

In the experiments, we choose $\alpha = 40(J^{(m)} - 1)$ and $\beta = 10$. With these chosen parameters, we are able to estimate the expected value for each element of $\boldsymbol{\eta}^{(mj)}$, which is given by $\mathbb{E}[\eta_j^{(mj)}] = 0.8$ and $\mathbb{E}[\eta_l^{(mj)}] = 0.2/(J_m - 1)$ for all $l \neq j$, satisfying $\mathbb{E}[\eta_j^{(mj)}] \gg \mathbb{E}[\eta_l^{(mj)}]$. We want to emphasize that these are the expected values of the prior distribution; and the posterior distribution for $\boldsymbol{\eta}$ is learned through training.

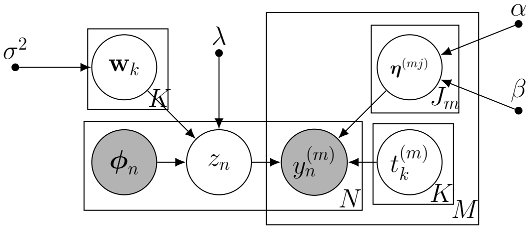


Figure 2: The graphical model.

2.5 Overall Model

We have described the proposed discriminative probability model. The joint distribution conditioned on the observed features $\boldsymbol{\Phi}$ is given by

$$p(\mathbf{Y}, \mathbf{W}, \mathbf{Z}, \mathbf{T}, \boldsymbol{\eta} | \boldsymbol{\Phi}) = \prod_{k=1}^{K} p(\mathbf{w}_k|\sigma^2) \prod_{m=1}^{M} \prod_{j=1}^{J_m} p\left(\boldsymbol{\eta}^{(mj)}|\alpha, \beta\right)$$
$$\prod_{m=1}^{M} \prod_{k=1}^{K} p\left(t_k^{(m)}\right) \prod_{n=1}^{N} p\left(z_n|\mathbf{W}, \boldsymbol{\phi}_n\right) \prod_{m=1}^{M} \prod_{n=1}^{N} p\left(y_n^{(m)}|z_n, \boldsymbol{\eta}, \mathbf{T}\right). \tag{11}$$

The proposed model is summarized using a graphical model in Figure 2.

3 Maximum a Posteriori Probability

In Section 2, we have presented our model. In this section, we introduce how we train the model. In this paper, we learn the maximum a posteriori probability (MAP) estimator for $\mathbf{W}$ through Expectation Maximization (EM) [21]. In the expectation step, we first compute the expected value of the logarithm of the joint distribution (i.e., the logarithm of Equation (11)) with respect to the posterior distribution $p(\mathbf{Z}, \mathbf{T}, \boldsymbol{\eta} | \boldsymbol{\Phi}, \mathbf{Y}, \widehat{\mathbf{W}})$, where $\widehat{\mathbf{W}}$ is the current estimate of $\mathbf{W}$. Then, in the maximization step, we update the estimate of $\mathbf{W}$ to maximize this expected value, i.e.,

$$\widehat{\mathbf{W}} = \arg\max_{\mathbf{W}} \mathbb{E}[\log p(\mathbf{Y}, \mathbf{W}, \mathbf{Z}, \mathbf{T}, \boldsymbol{\eta} | \boldsymbol{\Phi})] \tag{12}$$

We have already derived the objective function of the maximization step in Equation (4). We obtain the optimal $\widehat{\mathbf{W}}$ using conjugate gradient method [22].

We have derived the maximization step. However, we have a problem in the expectation step, because the posterior distribution $p(\mathbf{Z}, \mathbf{T}, \boldsymbol{\eta} | \boldsymbol{\Phi}, \mathbf{Y}, \widehat{\mathbf{W}})$ is computationally intractable. Therefore, we use a variational distribution $q(\mathbf{Z}, \mathbf{T}, \boldsymbol{\eta})$ to approximate it such that

$$q(\mathbf{Z}, \mathbf{T}, \boldsymbol{\eta}) \approx p(\mathbf{Z}, \mathbf{T}, \boldsymbol{\eta} | \boldsymbol{\Phi}, \mathbf{Y}, \widehat{\mathbf{W}}). \tag{13}$$

To ensure $q(\mathbf{Z}, \mathbf{T}, \boldsymbol{\eta})$ is tractable, we apply mean-field approximation such that

$$q(\mathbf{Z}, \mathbf{T}, \boldsymbol{\eta}) = \prod_{n=1}^{N} q(z_n) \prod_{m=1}^{M} \prod_{k=1}^{K} q\left(t_k^{(m)}\right) \prod_{m=1}^{M} \prod_{j=1}^{J_m} q\left(\boldsymbol{\eta}^{(mj)}\right). \tag{14}$$

In the inference, we derive an optimal variational distribution that minimizes the KL divergence between the variational distribution and the posterior distribution.

As described in [23], it is straightforward to derive the updates for the variational distribution by applying variational calculus. We omit the details of derivation, and list the update equations as follows:

$$q(z_n) \sim \text{Categorical}(\boldsymbol{\rho}_n) \tag{15}$$

where $\boldsymbol{\rho}_n$ is a k-dimensional vector, each element of which is given by

$$\begin{aligned}\rho_{nk} \propto \exp\Bigg\{ &\sum_{m=1}^{M}\sum_{j=1}^{J_k} \mathbb{E}_q[\mathbf{1}(t_k^{(m)} = j)] \sum_{l=1}^{J_k} \mathbf{1}(y_n^{(m)} = l)\mathbb{E}_q[\log \eta_l^{(mj)}] \\ &+ \widehat{\mathbf{w}}_k^T \boldsymbol{\phi}_n + \lambda_w \widehat{\mathbf{w}}_k^T \widehat{\mathbf{w}}_k - \log \sum_{i=1}^{K} \exp\left(\widehat{\mathbf{w}}_i^T \boldsymbol{\phi}_n + \lambda_w \widehat{\mathbf{w}}_i^T \widehat{\mathbf{w}}_i \right) \Bigg\}.\end{aligned} \tag{16}$$

$\boldsymbol{\rho}_n$ is normalized such that $\sum_{k=1}^{K} \rho_{nk} = 1$.

$$q\left(t_k^{(m)}\right) \sim \text{Categorical}\left(\boldsymbol{\zeta}^{(mk)}\right), \tag{17}$$

where $\boldsymbol{\zeta}^{(mk)}$ is a J_m dimensional vector, each element of which is given by

$$\zeta_j^{(mk)} \propto \exp\left\{ \textstyle\sum_{n=1}^{N} \mathbb{E}_q[\mathbf{1}(z_n = k)] \sum_{l=1}^{J_m} \mathbf{1}(y_n^{(m)} = l)\mathbb{E}_q[\log \eta_l^{(mj)}] \right\}, \tag{18}$$

$\boldsymbol{\zeta}^{(mk)}$ is normalized such that $\sum_{j=1}^{J_m} \zeta_j^{(mk)} = 1$.

$$q\left(\boldsymbol{\eta}^{(mj)}\right) \sim \text{Dirichlet}(\boldsymbol{\alpha}_{\boldsymbol{\eta}^{(mj)}}), \tag{19}$$

where $\boldsymbol{\alpha}_{\boldsymbol{\eta}^{(mj)}}$ is a J_m dimensional vector defined as

$$\alpha_{\boldsymbol{\eta}_l^{(mj)}} = \begin{cases} \alpha + \displaystyle\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{J_k} \mathbb{E}_q[\mathbf{1}(z_n = k)]\mathbb{E}_q[\mathbf{1}(t_k^{(m)} = j)]\mathbf{1}(y_n^{(m)} = l), \text{ if } l = j \\ \beta + \displaystyle\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{J_k} \mathbb{E}_q[\mathbf{1}(z_n = k)]\mathbb{E}_q[\mathbf{1}(t_k^{(m)} = j)]\mathbf{1}(y_n^{(m)} = l), \text{ if } l \neq j. \end{cases} \tag{20}$$

In the update equations, $\mathbb{E}_q$ represents that the expected value is taken with respect to the variational distribution q. The expected values involved are given as follows:

$$\mathbb{E}_q[\mathbf{1}(z_n = k)] = \rho_{nk}, \tag{21}$$

$$\mathbb{E}_q[\mathbf{1}(t_k^{(m)}) = j] = \zeta_j^{(mk)}, \tag{22}$$

$$\mathbb{E}_q[\log \eta_l^{(mj)}] = \psi\left(\alpha_{\eta_l^{(mj)}}\right) - \psi\left(\textstyle\sum_{i=1}^{J_m} \alpha_{\eta_i^{(mj)}}\right), \tag{23}$$

where ψ is the digamma function, i.e. the logarithmic derivative of the gamma function.

We summarize the training process in Algorithm 1.

Algorithm 1 Variational Expectation Maximization

```
repeat
    for $n \leftarrow 1$ to $N$ do
        update $q(\mathbf{z}_n)$ according to Equation (15).
    end for
    for $m \leftarrow 1$ to $M$ do
        for $k \leftarrow 1$ to $K$ do
            update $q(t_k^{(m)})$ according to Equation (17).
        end for
    end for
    for $m \leftarrow 1$ to $M$ do
        for $j \leftarrow 1$ to $J_m$ do
            update $q(\boldsymbol{\eta}^{(mj)})$ according to Equation (19).
        end for
    end for
    Update $\widehat{\mathbf{W}}$ based on the Equation (4) using conjugate
    gradient method.
until $\widehat{\mathbf{W}}$ converges
```

4 Experiments

In this section, we first present the experimental results on benchmark data. We demonstrate that the proposed method is able to learn the number of clusters and reveal the clustering structure by applying the method on benchmark data. Then we further illustrate the usefulness of the method with a real-world application.

4.1 Benchmark Data

We test the proposed method with 5 UCI data sets[24]: *iris* data set that collects 150 samples with 4 features from 3 different iris plants; *seeds* data set contains 210 samples described by 7 geometric parameters of kernels belonging to 3 varieties of wheats; *breast* data set contains the impedance measurements of 106 breast tissue samples from 6 classes; *glass* data set contains the 10 oxide content features of 214 glass samples from 6 types; *steel* data set contains 27 features of 1,941 samples of steel plates faults from 7 types. We also test on 3 face recognition data sets: *Yale* data set contains 165 face images of 32 × 32 pixels from 15 persons; *warpAR10P* data set contains 130 face images of 40 × 60 pixels from 10 persons; *warpPIE10P* data set contains 210 face images of 44 × 55 pixels from 10 persons.

For benchmark data, we only have access to the ground-truth cluster labels, but multi-expert labels are not available. Therefore, we generate labels for 10 synthetic experts, based on the ground-truth labels. For each expert, we first randomly partition the ground-truth cluster labels into 3 sets. For samples whose labels are in each of the 3 sets, we assume the expert gives positive labels, negative labels and decides to not provide a label, respectively. This simulates the situation that each expert is interested in one particular binary classification task related to the ground-truth clusters. The expert might be uncertain what label should be given for samples from certain ground-truth clusters, and decides to not provide a label. We randomly flip 10% of the labels to simulate the error of expert labels. We generate binary expert labels only, such that the ground-truth number of clusters is not obvious by observing the number of clusters from each expert.

In the experiments, we vary the percentage of the labels observed from each expert from 10% to 100%. To achieve this, we randomly pick a subset of labels from each expert independently. We conduct 5-fold cross validation, and measure the performance by comparing

Normalized Mutual Information (NMI) [11] between the learned cluster indicators and the ground-truth labels in both training and validation sets. NMI is the normalized version of mutual information such that it has a value between 0 and 1, where a larger value indicates a better performance.

Competing Methods

We compare the proposed methods with the following methods:

K-Means-Based Consensus Clustering (KCC) [15] is a cluster ensemble method that learns a median consensus clustering solution such that the similarity between consensus result and all given clustering solutions is maximized. This method uses partition expert labels only, without accessing the features. It can not directly predict out-of-bag validation samples. Therefore, we train a multinomial logistic regression using the features and learned cluster labels in the training set. Then, we predict the cluster assignment in the validation set using the trained logistic regression model. We denote this method using *KCC+LR*.

Metric Pairwise Constrained KMeans (MPCKMeans)[25] is a semi-supervised learning algorithm that combines constrained clustering and metric learning. We generate pairwise must link and cannot link between two samples, if 80% of synthetic experts agree that they should be in the same cluster and in different clusters, respectively. Since we include some high-dimensional data, we apply a scalable version that learns diagonal covariance matrices, ignoring the covariance between features.

Semi-crowdsourced Clustering (SemiCrowd)[8] is a crowd clustering method that first completes the similarity matrix via convex optimization and then learns a distance metric that makes use of observed features. It is not straightforward to predict out-of-bag samples with this model, and the matrix-completion optimization is not scalable. We are not able to apply this method to data sets with more than 500 samples.

Multi-Expert Constrained Clustering (MECC) [9] is a crowd clustering method that fit a multinomial logistic regression model to generate a clustering result that best predicts the observed pairwise similarity labels.

In addition, we apply *k-means* [26] as a baseline, which makes use of observed features only, without using the expert labels. We also include *Dirichlet Process Gaussian Mixture Model (DPGMM)* [19] because it automatically learns the number of clusters.

Experimental Results

We report NMI in training and validation sets in Figure 3. We observe from this figure that the proposed method is one of the best performers in terms of NMI in both training and validation set. Note that all other methods, except for DPGMM, are provided with the ground-truth number of clusters as a given parameter. The proposed method is at a disadvantage, because it automatically learns the number of clusters, and thus is provided with less ground-truth information.

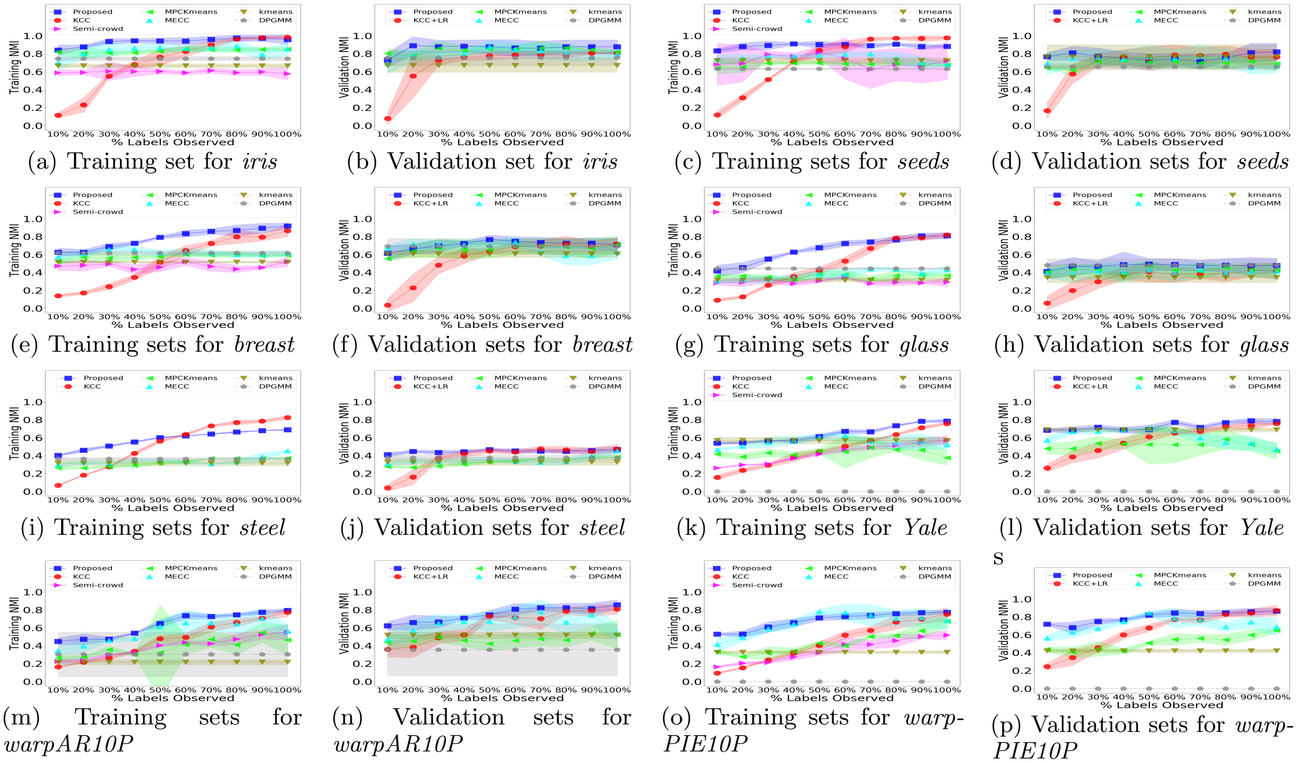
KCC performs badly when less labels are observed in the training data, because it only makes use of the observed expert labels without accessing the features. The proposed method is able to combine expert labels with observed features, which makes it perform better when less expert labels are given. In the training set of *seeds* and *steel*, KCC gives a higher NMI than the proposed method when more labels are observed. However, in the validation sets, KCC+LR does not outperform the proposed method. This suggests that when features are noisy, expert labels might be more trustworthy. The proposed method might be negatively influenced by the noisy features in the training results. However, because of the noisy features, KCC+LR does not generalized better in the validation set.

We can also observed in the figure that the pairwise similarity based methods, including MPCKMeans, semicrowd and MECC, usually perform worse. Note that in the experiments we convert the partition labels into pairwise similarity labels. This results in dense pairwise similarity matrices. These methods might not perform well on dense similarity matrices.

As mentioned previously, the proposed method is able to automatically determine the number of clusters. We summarize the mean and standard deviation of the number of clusters discovered by the proposed method for different tasks in Table 1. We also report the results of DPGMM for comparison. It can be concluded from the table that the proposed method is able to recover the number of clusters pretty accurately in most of the data sets. We also observed that, the proposed method overestimates the numbers of clusters for *steel* and *warpPIE10P* data set, probably because the features in these data sets are more noisy. Note that as shown in Figure 3, the proposed method still performs comparably with other methods on these two tasks in terms of NMI. DPGMM does not make use of the expert-label information, and performs worse. In *steel*, *Yale* and *warpPIE10P* data sets, DPGMM fails probably because each cluster in these data sets does not follow a Gaussian distribution.

4.2 COPD Application

The proposed method is developed to solve a real-world crowd-clustering problem, the Chronic Obstructive Pulmonary Disease (COPD) subtyping problem. COPD is a common lung disease related to cigarette smok-



(a) Training set for *iris* (b) Validation set for *iris* (c) Training sets for *seeds* (d) Validation sets for *seeds*

(e) Training sets for *breast* (f) Validation sets for *breast* (g) Training sets for *glass* (h) Validation sets for *glass*

(i) Training sets for *steel* (j) Validation sets for *steel* (k) Training sets for *Yale* (l) Validation sets for *Yale* s

(m) Training sets for *warpAR10P* (n) Validation sets for *warpAR10P* (o) Training sets for *warp-PIE10P* (p) Validation sets for *warp-PIE10P*

Figure 3: Normalized Mutual Information (NMI) in training and validation sets.

Table 1: Number of Clusters Discovered

data sets	Gound Truth	DPGMM	Proposed method with different precentage of labels observed					
			10%	30%	50%	70%	90%	100%
iris	3	3.4 (0.4)	3.0 (0.0)	3.0 (0.0)	3.2 (0.4)	3.4(0.5)	3.2 (0.4)	3.2 (0.4)
seeds	3	5.8 (0.4)	3.0 (0.0)	3.6 (0.5	3.8 (0.4)	4.4(0.5)	4.4 (0.5)	4.6 (0.5)
breast	6	10.6(1.7)	3.8 (0.4)	4.8 (0.4)	4.8 (0.4)	5.4 (0.8)	5.8 (0.4)	5.8 (0.4)
glass	6	16.8 (1.5)	4.6 (0.5)	7.0 (0.6)	6.4(1.5)	8.4 (0.8)	8 (0.8)	8.4 (0.8)
steel	7	1.8(1.9)	23.2 (2.8)	28.4(3.8)	31.0 (1.1)	31.4 (1.5)	30.4 (1.4)	30.0(2.8)
Yale	15	1.8 (0.4)	16.4(1.6)	14.6 (1.5)	13.2 (2.6)	14.6 (1.9)	16.4 (2.1)	15.8 (1.2)
warpAR10P	10	21.4 (11.1)	11.0 (1.9)	11.8 (2.8)	11.8 (1.7)	14.8(1.9)	13.2 (1.2)	14.2 (1.6)
warpPIE10P	10	1.2 (0.4)	20.2 (1.3)	17.2 (2.9)	14.8 (1.7)	16.4 (1.6)	16.0 (2.3)	19.4 (1.7)

ing. It is characterized by chronic, progressive, and irreversible lung airflow obstruction. It is predicted to be the third leading cause of death worldwide by the year of 2020 [27]. COPD is a clinically heterogeneous disease that may be separated into multiple subtypes (clusters) relevant to disease prognosis and treatment.

We try to discover the subtypes using a COPD data set. This data set contains 2, 109 subjects with 39 features. The subjects include heavy cigarette smokers with and without COPD. The features are collected based on clinical information, lung function, and measures from computed tomography (CT) chest imaging. We collected 63 clustering solutions provided by a cohort of COPD researchers, including pulmonologists, radiologists and data analysts. In each clustering solution, a subset (ranging from 85 to 2, 109 samples) of the subjects are partitioned into different number of clusters (ranging from 2 to 10).

We randomly split the data set into training and validation sets of equal sizes. We first train the proposed model using the training set. The proposed model finds 4 clusters in the data set. Note that since the number of clusters varies across different expert solutions, it is not easy to choose the number of clusters for the consensus result by hand. The proposed method is able to automatically determine the number of clusters, which is useful in this application.

We also train the competing methods to partition samples into 4 clusters. We predict clusters in the validation set using the trained models to check how well the learned clustering solution generalizes for out-of-bag samples. Since we do not have the ground-truth labels, we are not able to compute NMI. Rather, we check whether subjects in different clusters differ in terms the following 4 genetic risk scores [28, 29]: copdScore, lungfxScore, emphScore and airScore. These genetic risk scores measure the accumulation of genetic risk to different aspects of COPD and differences in genetic risk between COPD clusters may highlight biologic differences between clusters. Genetic risk score differences were evaluated via Kruskal-Wallis one-way analysis of variance [30], which is a non-parametric method for

Table 2: p-values in Kruskal-Wallis test

	copdScore	lungfxScore	emphScore	airScore
Proposed	**3.73E-07**	**1.52E-02**	**1.89E-04**	**1.18E-02**
KCC+LR	**1.14E-07**	3.61E-01	**2.44E-03**	**2.43E-02**
MPCKMeans	**1.96E-05**	9.92E-01	**6.64E-03**	7.81E-01
MECC	**7.45E-05**	6.27E-01	5.61E-02	6.01E-02
kmeans	3.38E-01	**3.29E-02**	7.07E-01	1.06E-01

testing whether samples in different groups originate from the same distribution. Note that the 39 features we used to train the models do not directly contain gene features. We summarize the p-value of Kruskal-Wallis test in Table 2.

In the table, we bold all p-values that are less than 0.05, which implies statistical significance. The proposed method is the only method that achieves statistical significance in all 4 genetic risk scores. This suggests that the proposed method outperforms other methods, in terms of discovering clusters that are more correlated with the COPD-relevant genetic information of the subjects.

In this application, we are not only interested in finding a clustering solution. We also want to understand how the expert labels are related, and what the experts agree or disagree with each other.

We first summarize the mean and standard deviation of some important features for each learned cluster in Table 3. In this table, we observe that Cluster 1 contains subjects that are more resistant to cigarette smoking, which is characterized by a high $FEV1pp\_utah$ value. Cluster 2 corresponds to airway disease predominant group, which is characterize by low $FEV1pp\_utah$, and high $WallAreaPct\_seg$. Cluster 3 corresponds to resistant cigarette smoker with mild emphysema, which is characterized by high $FEV1pp\_utah$ and mild $pctEmph$. Cluster 4 corresponds to the sickest subjects, with low $FEV1pp\_utah$, high $pctEmph$ and high $WallAreaPct\_seg$.

Now we analyze the relationship between the clustering results and expert labels. We pick the expert solutions that are accurately predicted based on learned clustering results, such that the prediction accuracy is above 80% in the training set, where the predicted labels are given by estimating $\mathbb{E}_q[p(y_n^{(m)}|z_n, \mathbf{T})]$, i.e., the expected value of Equation (8).

Then, we analyze $q(\mathbf{T})$ to observe the mapping between the learned clusters and the observed labels, where we use alphabet letters to represent the experts. The results are summarized using 6 groups, as shown in Table 4. We observe in the table that Groups 1 contains 9 solutions. These solutions agree to separate the cluster 4 from the the rest, i.e., they separate the sickest subjects from the healthier subjects. Group 2 agrees with Group 1, but with positive and negative labels flipped. Group 3 separates the airway disease predominant group from the rest. Both Groups 4 and 5 separate clusters 1 and 3 from clusters 2 and 4, with labels flipped. They separate the more resistant cigarette smokers from the sicker subjects. Expert O in fact partition the subjects into 5 groups, but label 2 and label 4 are not matched by any learned clusters. Labels 0, 1, and 3 in this solution corresponds to resistant cigarette smokers, airway disease predominant subjects and the sickest subjects, respectively.

Table 3: Clustering Results

	No. samples	FEV1pp_utah	pctEmph	WallAreaPct_seg	Emph_UL_LL_ratio
1	382	93.6 (14.6)	1.8 (1.7)	59.9 (2.5)	1.6 (1.2)
2	137	65.7 (16.9)	1.9 (1.9)	65.4 (2.3)	2.3 (2.2)
3	139	91.0 (13.5)	7.6 (4.0)	59.1 (2.3)	2.4 (3.6)
4	311	44.9 (18.6)	22.2 (12.1)	62.4 (2.7)	2.0 (2.3)

Table 4: Mapping between Clusters and Expert Labels

	Solutions	Mapping
1	A, B, C, D, E F, G, H, I	Label 0: Cluster 1, 2 & 3 Label 1: Cluster 4
2	J	Label 0: Cluster 4 Label 1: Cluster 1 , 2 & 3
3	K	Label 0: Cluster 2 Label 1: Cluster 1 , 3 & 4
4	L, M	Label 0: Cluster 1 & 3 Label 1: Cluster 2 & 4
5	N	Label 0: Cluster 2 & 4 Label 1: Cluster 1 & 3
6	O	Label 0: Cluster 1 & 3 Label 1: Cluster 2 Label 3: Cluster 4

As shown in this table, the proposed method helps us better understand the expert labels. Existing crowdclustering methods analyze the pairwise similarity labels and it would be more difficult to reveal such relationship between expert solutions.

5 Conclusion

In this paper, we proposed a crowdclustering model that directly analyzes partition labels. The proposed model adopts a novel approach to generate latent cluster indicators, such that it simultaneously determines the number of clusters and partitions samples into clusters. The proposed model also learns a mapping between the latent clusters and expert labels, revealing the relationships between labels from different experts. Experiments on benchmark data demonstrates that the proposed model simultaneously learns the number of clusters and discovers the clustering structure. An experiment on a real-world disease subtyping problem illustrates that the proposed model helps us understand the agreement and disagreement between experts.

Acknowldgement

We thank the COPDGene Team and the support from the National Heart, Lung, and Blood Institute, Award Number U01 HL089856. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Heart, Lung, and Blood Institute or the National Institutes of Health.

References

[1] D. Niu, J. G. Dy, and Z. Ghahramani, "A nonparametric bayesian model for multiple clustering with overlapping feature views," in *The fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, 2012, pp. 814–822.

[2] D. Niu, J. G. Dy, and M. Jordan, "Iterative discovery of multiple alternativeclustering views," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 7, pp. 1340–1353, 2014.

[3] D. Klein, S. D. Kamvar, and C. D. Manning, "From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering," 2002.

[4] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *the Eighteenth International Conference on Machine Learning (ICML 2001)*, vol. 1, 2001, pp. 577–584.

[5] E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, 2002, pp. 505–512.

[6] R. G. Gomes, P. Welinder, A. Krause, and P. Perona, "Crowdclustering," in *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, 2011, pp. 558–566.

[7] J. Yi, R. Jin, A. K. Jain, and S. Jain, "Crowdclustering with sparse pairwise labels: A matrix completion approach," in *AAAI Workshop on Human Computation*, vol. 2, 2012.

[8] J. Yi, R. Jin, S. Jain, T. Yang, and A. K. Jain, "Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning," in *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012, pp. 1772–1780.

[9] Y. Chang, J. Chen, M. Cho, P. Castaldi, E. Silverman, and J. Dy, "Clustering from multiple uncertain experts," in *20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*, 2017, pp. 28–36.

[10] Y. Chang, J. Chen, M. H. Cho, P. J. Castaldi, E. K. Silverman, and J. G. Dy, "Multiple clustering views from multiple uncertain experts," in *The 34th International Conference on Machine Learning (ICML 2017)*, 2017, pp. 674–683.

[11] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *Journal of machine learning research*, vol. 3, no. Dec, pp. 583–617, 2002.

[12] A. Topchy, A. K. Jain, and W. Punch, "Combining multiple weak clusterings," in *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on.* IEEE, 2003, pp. 331–338.

[13] A. Topchy, A. K. Jain and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.

[14] A. L. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[15] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen, "K-means-based consensus clustering: A unified view," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 1, pp. 155–169, 2015.

[16] S. Vega-Pons and J. Ruiz-Shulcloper, "A survey of clustering ensemble algorithms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 03, pp. 337–372, 2011.

[17] E. Simpson, S. J. Roberts, A. Smith, and C. Lintott, "Bayesian combination of multiple, imperfect classifiers," 2011.

[18] H.-C. Kim and Z. Ghahramani, "Bayesian classifier combination," in *the 15th Artificial Intelligence and Statistics (AISTATS 2012)*, 2012, pp. 619–627.

[19] C. E. Antoniak, "Mixtures of dirichlet processes with applications to bayesian nonparametric problems," *The annals of statistics*, pp. 1152–1174, 1974.

[20] D. M. Blei, M. I. Jordan *et al.*, "Variational inference for dirichlet process mixtures," *Bayesian analysis*, vol. 1, no. 1, pp. 121–143, 2006.

[21] 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. Series B (methodological)*, pp. 1–38, 1977.

[22] J. R. Shewchuk *et al.*, "An introduction to the conjugate gradient method without the agonizing pain," 1994.

[23] C. M. Bishop *et al.*, *Pattern recognition and machine learning.* Springer, New York, 2006, vol. 1, ch. 10 Approximate Inference, pp. 461 – 474.

[24] M. Lichman, "UCI machine learning repository," http://archive.ics.uci.edu/ml, 2013.

[25] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of the twenty-first international conference on Machine learning (ICML 2004).* ACM, 2004, p. 11.

[26] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[27] J. Vestbo, S. S. Hurd, A. G. Agustí, P. W. Jones, C. Vogelmeier, A. Anzueto, P. J. Barnes, L. M. Fabbri, F. J. Martinez, M. Nishimura *et al.*, "Global strategy for the diagnosis, management, and prevention of chronic obstructive pulmonary disease: Gold executive summary," *American journal of respiratory and critical care medicine*, vol. 187, no. 4, pp. 347–365, 2013.

[28] M. H. Cho, P. J. Castaldi, C. P. Hersh, B. D. Hobbs, R. G. Barr, R. Tal-Singer, P. Bakke, A. Gulsvik, R. San José Estépar, E. J. Van Beek *et al.*, "A genome-wide association study of emphysema and airway quantitative imaging phenotypes," *American journal of respiratory and critical care medicine*, vol. 192, no. 5, pp. 559–569, 2015.

[29] J. D. Morrow, M. H. Cho, C. P. Hersh, V. Pinto-Plata, B. Celli, N. Marchetti, G. Criner, R. Bueno, G. Washko, K. Glass *et al.*, "Dna methylation profiling in human lung tissue identifies genes associated with copd," *Epigenetics*, vol. 11, no. 10, pp. 730–739, 2016.

[30] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American statistical Association*, vol. 47, no. 260, pp. 583–621, 1952.