Mixed Membership Subspace Clustering

Stephan Günnemann
Carnegie Mellon University, USA
sguennem@cs.cmu.edu

Christos Faloutsos
Carnegie Mellon University, USA
christos@cs.cmu.edu

Abstract—Clustering is one of the fundamental data mining tasks. While traditional clustering techniques assign each object to a single cluster only, in many applications it has been observed that objects might belong to multiple clusters with different degrees. In this work, we present a Bayesian framework to tackle the challenge of mixed membership clustering for vector data. We exploit the ideas of subspace clustering where the relevance of dimensions might be different for each cluster. Combining the relevance of the dimensions with the cluster membership degree of the objects, we propose a novel type of mixture model able to represent data containing mixed membership subspace clusters. For learning our model, we develop an efficient algorithm based on variational inference allowing easy parallelization. In our empirical study on synthetic and real data we show the strengths of our novel clustering technique.

I. INTRODUCTION

Clustering is an established mining task for grouping objects based on their mutual similarity. Traditional clustering approaches focus on finding disjoint clusters. In many applications, however, it is observed that objects naturally belong to multiple clusters with different degrees. In customer segmentation, for example, a person might primarily show the characteristics of an engineer (the customer’s occupation) and simultaneously some aspects of a soccer fan (the customer’s hobby). Other domains include movie databases, where movies belong to multiple genres with different strengths, bibliographic databases, where authors can be grouped according to multiple research topics, and gene data, representing the genes’ multiple functions.

Exploiting the model-based clustering paradigm, mixed membership models solve the task of assigning objects to multiple clusters with different degrees. While these models are well established in the text mining area (e.g. LDA [1]), far less approaches have been introduced for vector data (cf. Section IV).

It is worth mentioning that the concept of mixed membership is conceptually very different from uncertain membership (cf. [2], [3]). The latter one is used, e.g., ‘traditional’ mixture models where one models the uncertainty of assigning an object to a specific cluster. Removing the uncertainty, each object belongs to a single cluster only. In contrast, in mixed membership clustering, an object might belong to multiple clusters even when considering a deterministic setting. Or as intuitively stated in [2]: “Being certain that a person is partly Asian and partly European, is very different than being uncertain about a person’s ethnic background. More information about the person, such as DNA tests, could resolve uncertainty, but cannot make the person change his ethnic membership”. Overall, the principle of soft clustering as exploited in ‘traditional’ mixture models does not support the idea of mixed memberships.

In this work, we tackle the challenge of mixed-membership clustering for vector data by combining it with the ideas of subspace clustering [4]. In contrast to full-space clustering where each dimension is regarded as equally important, subspace clustering allows varying importance of the dimensions – individually for each cluster. Thus, the dimension ‘income’, e.g., might be highly relevant for the cluster ‘engineer’ but less relevant for the ‘soccer fan’. The dimension ‘time spend outside’, in contrast, might be less important for the ‘engineer’. Consequently, when a person belongs to both clusters, it is intuitive to assume that the person’s income is very likely due to the ‘engineer’ cluster, while the ‘time spend outside’ is due to the other cluster. Overall, based on the degree an object belongs to a specific cluster and based on the relevance of a dimension for this cluster, one can determine which cluster is responsible for the object’s attribute value in the currently considered dimension.

Based on this general idea, we present a Bayesian framework that models data containing mixed membership subspace clusters. The contributions of our work are:

- We exploit the model-based clustering paradigm to solve the task of mixed membership clustering for vector data: each object can belong to multiple clusters with different degrees.
- Our model represents the locally varying relevance of dimensions associated to the clusters, thus, adapting ideas from the subspace clustering paradigm.
- We propose an efficient learning algorithm based on the principle of variational inference.

Since the terms ‘subspace clustering’, ‘co-clustering’, and ‘biclustering’ are ambiguously used in the literature, we kindly refer to Section IV for clarification. In particular, we want to mention that the methods [5], [6], [7] solve a completely different task than our method.

The paper is structured as follows: Sec.II describes our Bayesian framework for mixed membership subspace clustering. We propose an efficient learning algorithm in Sec.III. Related work is described in Sec.IV. We present experimental results in Sec.V and we conclude in Sec.VI.
II. BAYESIAN FRAMEWORK

In this section, we introduce our Bayesian framework that models data containing mixed membership subspace clusters. Before presenting the details of our model, we start with a (simplified) example describing the general idea.

**Introductory example.** Consider a 5-dimensional data-space containing 3 subspace clusters. Let us assume the subspace clusters follow normal distributions with means as illustrated in Figure 1 (in this example, we ignore the variance for simplicity). Since we consider subspace clusters, some dimensions might be irrelevant for individual clusters. In the figure, we illustrate the relevance via colors: the lighter the color, the less relevant the dimension for the corresponding cluster. Thus, dimension 4 and 5, for example, are mostly irrelevant for cluster 1. In our model, we will model the relevance via general weight vectors.

Given this setting, how will a dataset following this clustering structure look like? Since we consider mixed membership clustering, the answer depends on the membership information of the objects. Object $o_1$ in Fig. 1, e.g., belongs to cluster 1 and 2. Since cluster 1 is highly relevant in dimensions 1 and 2 (and cluster 2 irrelevant), the attribute values of object $o_1$ in dimension 1 and 2 should follow the distribution of cluster 1. Accordingly, it is easy to see that the object $o_1$ should follow cluster 2 in dimensions 4 and 5.

Since dimension 3 is relevant for both clusters, we have to be more careful: In general, when analyzing dimension $d$ and object $n$, we have to consider the degree the object belongs to a specific cluster and the relevance of this dimension for the cluster. Thus, assuming that dimension 3 is equally important for cluster 1 and 2, the object $o_1$ has a higher likelihood to follow cluster 1 in this dimension. In contrast, the object $o_2$ will more likely follow cluster 2 in dimension 3 since its membership degree for cluster 2 is higher. Note that the relevance of the dimension is important, too: Assuming that dimension 3 is far more relevant for cluster 3 than for cluster 1, the object $o_3$ should follow cluster 3 in this dimension.

Using this procedure, we can easily generate data containing mixed membership subspace clusters. It is worth mentioning that an object can belong to multiple clusters even when the clusters have ‘overlapping’ relevant dimensions. In the following, we will formalize this generative process based on a Bayesian framework, which subsequently allows us to learn the clustering structure in a probabilistic fashion.

**Formal Model.** An overview of our framework is given by the graphical model in Fig. 2. Following convention, we do not distinguish between a random variable $\mathbf{X}$ and its realization $\mathbf{x}$ if it is clear from the context. As an abbreviation, we denote sets of random variables with the index $*$, e.g. $Y_{*,d}$ is the set of all objects in dimension $d$. Vectors of random variables are written in bold font, e.g. $\mathbf{\pi}$, while the entries of the vectors are given in standard font, e.g. $\pi_k$.

The number of objects is denoted with $N$, the number of dimensions with $D$, and the number of clusters/components with $K$. We write $k \in K$, as a shortcut for $k \in \{1, \ldots, K\}$.

**Recap: Mixture Models.** In traditional mixture models, the probability density function when considering an object $X^{(n)}$ is given by $p(X^{(n)} \mid \pi, \Theta) = \sum_{k=1}^{K} \pi_k \cdot p_k(X^{(n)} \mid \Theta_k)$ where $\pi$ is a vector representing the mixing weights of the components, $p_k$ is the density function of the $k$-th component (e.g. a Normal distribution) and $\Theta_k$ are the corresponding parameters (e.g. mean and variance values).

From a generative perspective, data is generated in two steps: First, one samples for each object $n$ a component the object belongs to, i.e. one considers the latent variable $z_n \sim \text{Categorical}(\pi)$. Second, the actual data vector of object $n$ is generated by sampling from the $z_n$-th component, i.e. $X^{(n)}|z_n \sim p_{z_n}(\cdot \mid \Theta_{z_n})$.

Two aspects prevent the application of this model for mixed membership clustering. First, the mixing weights $\pi$ are identical for each object, i.e. the probability to be generated by a specific component is the same for each object. Second, the latent variable $z_n$ represents only a single value, thus the feature vector of each object is generated by a single component only. In the following, we extend this model to handle mixed membership subspace clusters.

**Locally Varying Relevance of Clusters.** The first issue is easy to solve by assigning an individual mixture $\pi^{(n)}$ to each object $n$. As established [1], [5], [6], these weights can be modeled by using a Dirichlet distribution, i.e.

$$\pi^{(n)} \sim \text{Dir}(\alpha) \quad (\text{M1})$$

where $\alpha$ is a $K$-dimensional vector representing the concentration parameters. The higher the value of $\pi_k^{(n)}$, the higher the membership degree of object $n$ to cluster $k$. That is, the vector $\pi^{(n)}$ exactly reflects the membership information as illustrated in Figure 1. By using the Dirichlet distribution we ensure that $\sum_{k} \pi_k^{(n)} = 1$.

Additionally, since we follow the idea of subspace clustering where the relevance of dimensions might be different for each cluster, we also introduce for each dimension an individual relevance distribution over the set of clusters:

$$w^{(d)}_k \sim \text{Dir}(\beta^{(d)}) \quad (\text{M2})$$

The value $w^{(d)}_k$ represents the relevance of cluster $k$ in dimension $d$. Two crucial aspects have to be mentioned:

First, the weights $w^{(d)}_k$ are normalized for each dimension individually. That is, it holds $\sum_{k} w^{(d)}_k = 1$. This way, it is possible to compare the relative importance of clusters in a

![Figure 1](image-url)
specific dimension. The relevance of the dimension 'income' for the cluster 'engineer', for example, might be twice as high as the relevance of this dimension for the cluster 'soccer fan'. It is not useful to normalize the weights for each cluster, i.e. requiring $\sum_d w_k^{(d)} = 1$. In this case, clusters covering many dimensions would be penalized. For example, a cluster covering all $D$ dimension with equal importance would lead to weights of $1/D$, while a cluster with only two relevant dimensions would get the higher weight $1/2$. By normalizing per dimension we are not biased to specific dimensionalities leading to a fair comparison between the clusters.

Second, for each dimension a different Dirichlet distribution is used based on the parameter $\beta^{(d)}$. We argue that the clusters' relevance might highly vary when considering different dimensions: just because a cluster has been relevant in, e.g., the first ten dimensions does not allow a conclusion about its relevance in the eleventh dimension. Thus, in our model (and also in traditional mixture models), dimensions and objects are explicitly handled differently, while methods as, e.g., [5], [6], [7] treat objects and dimensions (or generally: rows and columns of the data matrix) interchangeably.

**Interpretation as Conditional Marginals.** The second issue of traditional mixture mixture, i.e. the single variable $z_{n,d}$, is solved by introducing for each dimension an individual random variable $z_{n,d}$. In Figure 1, the object $o_1$, e.g., 'follows' cluster 1 in dimensions 1-3 but cluster 2 in dimensions 4 and 5. Though, what is the distribution of the variable $z_{n,d}$?

Intuitively, if $\pi_k^{(n)}$ is large, object $n$ should follow cluster $k$ in many of its dimensions. Simultaneously, if a cluster is highly relevant in a specific dimension, the probably that objects follow this cluster should be high. Based on this idea, we now present a principled solution how to derive the random variable $z_{n,d}$.

Let us consider the (unknown) 3-dimensional probability density function $g(N, D, K)$ denoting the probability to select an object $n$, a dimension $d$, and a cluster $k$. We see that the conditional distribution $g(K|N=n, D=d)$ reflects the probability to select a specific cluster given the object $n$ and the dimensions $d$, i.e. this conditional distribution exactly corresponds to $z_{n,d}: z_{n,d} \sim g(K|N=n, D=d)$.

At the same time, using the intuition from above, the term $\pi_k^{(n)}$ is reflected by the conditional, marginal distribution $g(K|N=n) = \sum_d p(D=d | N=n) \cdot g(K|N=n, D=d)$. If an object $n$ has a high value for $\pi_k^{(n)}$ the cluster $k$ should appear often when averaged over all dimensions. That is, we have $g(K=k|N=n) = \pi_k^{(n)}$. Accordingly, considering the term $w_k^{(d)}$, it is reflected by the conditional, marginal distribution of $K$ given $D$, i.e. $g(K=k|D=d) = w_k^{(d)}$.

The above information can be used to derive the value of $z_{n,d}$. First, following the ideas of [7], [5], [6], it is natural to assume that $N$ and $D$ are conditional independent given $K$. That is, given a specific cluster, the set of objects and the set of dimensions belonging to the cluster can be generated independently. Please note that this does not mean that objects and dimensions are clustered independently. Using this property, we derive

$$g(K|N, D) = \frac{g(N, D|K) \cdot g(K)}{g(N, D)} = \frac{g(N|K) \cdot g(D|K) \cdot g(K)}{g(N, D)} = \frac{g(N|K) \cdot g(D|K)}{g(N)}$$

Given $n$ and $d$, the term $\frac{g(N|K) \cdot g(D|K)}{g(N)}$ is constant. Thus,

$$p(z_{n,d} = k) = g(K = k | N = n, D = d) \propto \frac{\pi_k^{(n)} \cdot w_k^{(d)}}{g(K = k)}$$

The marginal distribution $g(K = k)$ intuitively represents the weight of a cluster independent of a specific object and dimension. It relates to the mixing weights as used in traditional mixture models. We introduce an additional $K$-dimensional random variable modeling these weights:

$$m \sim Dir(\xi) \quad (M3)$$

Using the fact $m_k = g(K = k)$, we can finally derive

$$z_{n,d} \sim \text{Categorical}(t^{(n,d)}) \quad , \quad t^{(n,d)} \sim \frac{\pi_k^{(n)} \cdot w_k^{(d)}}{m_k} \quad (M4)$$

Figure 3 shows an example how the distribution of $z_{n,d}$ is determined. The object 1 will most likely follow cluster 1 in dimension 1 (probability of 0.81) since both the object’s membership to this cluster (0.5) and the relevance of this cluster in this dimension (0.7) are high. In contrast, in dimension 2, the object will most likely follow cluster 3 since $P(z_{1,2}=3) \approx 0.72 \propto 0.4 \cdot 0.5 \cdot 0.3$ Overall, the formula naturally reflects the intuition which cluster is responsible for an object’s attribute value in a specific dimension. The more relevant a cluster in a specific dimension and the higher its membership degree for the current object, the more likely is the cluster selected. The division by $m_k$ ensures that clusters which are globally not very dominant, nonetheless will get the chance to be selected locally. Thus, the local membership degree and the global weight of a cluster are decoupled.
**Coupling of Dirichlet Parameters.** Interpreting the three variables $\pi$, $w$, and $m$ as (conditional) marginals, we see that they should not be unrelated to each other. E.g., assuming that for each object the cluster $k$ has very low appearance probability, then consequently also in vector $m$ its value needs to be low. We solve this issue by coupling the parameters of the corresponding Dirichlet distributions. The general idea is: If the variables really where true marginals, then $m \propto \sum_n \pi^{(n)}$ and $m \propto \sum_d w^{(d)}$ has to hold. Based on the summation rule of the Dirichlet distribution (if $X \sim Dir(\alpha)$ and $Y \sim Dir(\beta)$ then $X + Y \sim Dir(\alpha + \beta)$), we can therefore derive that

$$\xi = N \cdot \alpha \text{ and } \xi = \sum_d \beta^{(d)} \Rightarrow \alpha = N^{-1} \sum_d \beta^{(d)} \quad \text{(M5)}$$

Thus, the three variables are coupled via their corresponding Dirichlet parameters. Indeed, the parameters $\alpha$ and $\xi$ vanish since they are completely determined by the values $\beta^{(d)}$.

By coupling $\pi$, $w$, and $m$, the distributions match to each other. Though, theoretically it might happen that $\pi_k^{(n)} \cdot w_k^{(d)}$ becomes zero for each $k$. Meaning that an object $n$ only belongs to clusters which are completely irrelevant in dimension $d$. To handle this case we can follow the ideas of [7], [8] and assign the object to a designated noise/default component (e.g. denoted with the index K+1)

$$z_{n,d} = K+1 \quad \text{if } \forall k: \pi_k^{(n)} \cdot w_k^{(d)} = 0$$

Note that w.l.o.g. we assume $m_k \neq 0$ for all $k$. If a cluster would be completely empty, the assumption that the data is generated by $K$ components is obviously wrong and the empty component can simply be ignored. That is, in k-means, we continue our method with $K-1$ components.

**Final Feature Vectors.** Having determined the random variables $z_{n,s}$ according to Eq. M4 we can finally generate the feature vector $X^{(n)}$ of an object $n$. Informally, this involves collecting all dimensions $d$ referring to cluster $k$ and then using the density function $p_k$ of the $k$th component to generate the attribute values in the corresponding subspace. Formally, since $p_k$ can be any $D$-dimensional density function, we have to consider its marginal distribution in the corresponding subspace. W.l.o.g. let $\{1, ..., m\}$ denote the dimensions of object $n$ having selected cluster $k$ (i.e. $z_{n,i}=k$ for all $i \leq m$ and $z_{n,i} \neq k$ for $i > m$), then the attribute values of $X^{(n)}$ in subspace $\{1,...,m\}$ are distributed as

$$X_1^{(n)}, \ldots, X_m^{(n)} \sim \int \ldots \int p_k(X^{(n)} | \Theta_k) \quad \text{(M6)}$$

Simply stating: To generate the attribute values of an object in the subspace $\{1, ..., m\}$, we just have to sample from the $k$-th component a (full-dimensional) feature vector $Y$ and retain only its entries in the dimensions $\{1, ..., m\}$. The attribute values in the remaining dimensions are determined by sampling from the other components. It is also easy to incorporate missing values in this generative process by simply not generating some attribute values.

**Prior Distributions.** To realize a fully Bayesian model, the parameters $\Theta_k$ and $\beta^{(d)}$ can be equipped with appropriate prior distributions. The prior of $\Theta_k$ depends on the choice of the underlying component’s distribution $p_k$. Note that similar to traditional mixture models, we assumed that each object follows in each dimension a specific cluster. In some applications, however, it might be useful to consider noisy objects. This prior knowledge can be incorporated in our model by adapting the priors of the components, e.g., adding a bias to large variances for one of the components.

For the parameters $\beta^{(d)}$, we choose an exponentially distributed prior:

$$\sum_d \beta_k^{(d)} \sim Exp(\lambda) \quad \text{(M7)}$$

The motivation for this choice is: If the data inhibits a good clustering structure, we expect each object to belong to a few clusters only (if each object would belong to each cluster, a clustering is rather meaningless). Thus, the membership vectors $\pi$ should tend to be sparse. For Dirichlet distributions this aspect is realized by selecting small concentration parameters, i.e., $\sum_k \alpha_k$ should be small. Using Eq. M5, it follows that $\sum_k \beta_k^{(d)}$ should be biased to low values, which is realized by selecting the above prior.

**III. THE LEARNING ALGORITHM**

While the previous section focused on the model’s generative process, i.e., how to generate data containing mixed membership subspace clusters, we now present our learning technique. That is, given a set of observations $X$ we infer the model’s parameters. Due to space limitations, we will only present the most important results. Details are available on http://www.cs.cmu.edu/~sguennem/MMSC.pdf.

**A. Independent Gamma Distributions**

Our model uses the Dirichlet distributions $\pi^{(n)}$, $w^{(d)}$, and $m$. Since these distributions require a normalization, e.g. $\sum_k \pi_k^{(n)} = 1$, a learning is challenging. As a solution, we exploit a transformation to independent Gamma distributions [9]: Let $\tilde{\pi}_k \sim \text{Gamma}(\alpha_k,1)$ be a set of independent Gamma distributions with (potentially different) shape parameters $\alpha_k$ and let $\pi_k = \frac{\tilde{\pi}_k}{\sum_j \tilde{\pi}_j}$, then $(\pi_1, \ldots, \pi_K) \sim Dir((\alpha_1, \ldots, \alpha_K))$. Thus, the Dirichlet distribution can be constructed by normalizing independent Gamma distributions. Let $\tilde{\pi}_k^{(n)}$, $\tilde{w}_k^{(d)}$, $\tilde{m}_k$ be the Gamma distributions generating the Dirichlets $\pi^{(n)}$, $w^{(d)}$, and $m$. Using the above result in Eq. M4 leads to

$$f_k^{(n,d)} \sim \frac{\pi_k^{(n)} - w_k^{(d)}}{m_k} \quad \text{and} \quad \pi_k^{(n)} = \frac{\sum_j \pi_j^{(n)}}{\sum_j m_j^{(n)}} = \frac{\pi_k^{(n)} - w_k^{(d)}}{m_k} \quad \text{and} \quad \pi_k^{(n)} = \frac{\sum_j \pi_j^{(n)}}{\sum_j m_j^{(n)}}$$

It turns out that the categorical distribution of $z_{n,d}$ can directly be obtained through the Gamma distributions. We do not need to determine the normalized Dirichlets. Thus,
in the following we only need to consider the independent, Gamma distributed variables $\tilde{w}^{(n)}_k$, $\tilde{m}^{(d)}_k$, and $\tilde{m}_k$.

B. Variational Inference

The general inference problem we have to solve is to determine the distribution $p(V|X)$, where $V=\{\beta, \pi, m, w, z, \Theta\}$ is the set of all latent variables. Based on this distribution, we can, e.g., pick the realizations of the latent variables leading to the highest likelihood given the data. Computing $p(V|X)$, however, is intractable. Thus, we compute an approximation based on the principle of variational inference [10]: we approximate $p(V|X)$ by a tractable family of parameterized distributions $q(V|\Psi)$. The parameters $\Psi$ are the free variational parameters. These parameters are optimized such that the best approximation between $q$ and $p$ is obtained. Technically, one minimizes the Kullback-Leibler divergence between $q$ and $p$ by optimizing $\Psi$. Using Jensen’s inequality, minimizing the KL divergence is equivalent to maximizing the following lower bound on the log marginal likelihood [10]:

$$
\mathcal{L}(X;\Psi) = \mathbb{E}_q[\ln p(X|V)] - \mathbb{E}_q[\ln q(V|\Psi)]
$$  \hfill (1)

where $\mathbb{E}_q[.]$ denotes the expectation w.r.t. the $q$ distribution. We exploit the idea of mean field approximation, i.e. the function $q$ is assumed to factorize in

$$
p(V|X) \approx q(V|\Psi) := \prod_{d} \prod_{k} q_1(\tilde{w}^{(d)}_k) \cdot \prod_{n} \prod_{k} q_2(\tilde{m}^{(n)}_k) \cdot \prod_{k} q_3(\tilde{m}_k) \cdot q_4(\beta^*) \cdot \prod_{n} q_5(z_{n,d}) \cdot \prod_{k} q_6(\Theta_k)
$$

and we use the following variational distributions

$$
q_1(\tilde{w}^{(d)}_k) = \text{Gamma}(\tilde{w}^{(d)}_k|\alpha_d^{w},\beta_d^{w})
$$

$$
q_2(\tilde{m}^{(n)}_k) = \text{Gamma}(\tilde{m}^{(n)}_k|\alpha_n^{m},\beta_n^{m})
$$

$$
q_3(\tilde{m}_k) = \text{Gamma}(\tilde{m}_k|\alpha_k^{m},\beta_k^{m},\alpha_k^{m} > 1)
$$

$$
q_4(\beta^*) = \delta_{\beta^*}\quad \text{a priori: } \beta^{d,k} \in D, k \in K
$$

$$
q_5(z_{n,d}) = \text{Categorical}(z_{n,d} \mid \phi_{n,d,1},..,\phi_{n,d,K})
$$

$$
q_6(\Theta_k) = \delta_{\Theta_k}
$$

where $\Psi = \{\alpha_n^{w,*}, \beta_n^{w,*}, \alpha_n^{m,*}, \beta_n^{m,*}, \alpha_k^{m,*}, \beta_k^{m,*}, \phi_{n,d,*}, \phi_{n,d,K,*}, \Theta_k\}$ are the variational parameters to be optimized. Note that each distribution has its own variational parameters [10].

Thus, the functions $q_1(\tilde{w}^{(d)}_k)$ and $q_1(\tilde{w}^{(d)}_k)$, for example, are not necessarily identical. This extra degree of freedom allows to find a good approximation between $q$ and $p$.

The choice for the distributions of $q_1$, $q_2$ and $q_3$ follows from the properties of variational inference and mean field approximation [10], i.e. these are the optimal distributions given the above factorization. The function $q_5$ is selected to match with its related distributions of $q_1$ and $q_2$. The Dirac delta functions for $q_4$ and $q_6$ are selected for tractability since in this case we only need to compute point estimates.

General Processing Scheme. We use an iterative coordinate ascent method to maximize Equation 1 w.r.t. the parameters $\Psi$ (the update equations are presented in the Section III-C). The processing schema is as follows:

while $\Psi$ has not converged do

parallel for $d \in D, k \in K$ do update $\alpha_d^{w,k}, \beta_d^{w,k}$  \hfill Eq. U1

parallel for $n \in N, k \in K$ do update $\alpha_n^{m,k}, \beta_n^{m,k}$  \hfill Eq. U3

parallel for $k \in K$ do update $\alpha_k^{m,k}$  \hfill Eq. U4, U5

parallel for $d \in D$ do update $\beta_d^{w,k}$  \hfill Eq. U6

parallel for $n \in N, d \in D$ do update $\phi_{n,d,*}$  \hfill Eq. U7

parallel for $k \in K$ do update $\Theta_k$  \hfill Eq. U8, U9

We want to highlight that the update of $\alpha_d^{w,k}, \beta_d^{w,k}$ is independent of the one for $\alpha_n^{m,k}, \beta_n^{m,k}$. That is, the update of these parameters can be done in parallel for each $d \in D$ and $k \in K$. The same holds for the other steps of the algorithm. Note that due to the properties of variational inference [10], it is guaranteed that the method converges.

C. Update Equations

We briefly present the update equations required for the coordinate ascent method.

Update of $q_1(\tilde{w}^{(d)}_k)$. Following the principle of [10], the optimal distribution for $q_1(\tilde{w}^{(d)}_k)$ can be determined by

$$
\ln q_1^*(\tilde{w}^{(d)}_k) = \mathbb{E}_q[\ln p(X)] - \mathbb{E}_q[\ln q(V|\Psi)] + \text{const}
$$

Here, the constant $\text{const}$ absorbs all terms which are independent of $\tilde{w}^{(d)}_k$ and, thus, do not affect the optimal distribution of $q_1$. The term $\mathbb{E}_q[\ln q(V|\Psi)]$ denotes the expectation with respect to the distribution $q$ taken overall all variables $V$ except of $\tilde{w}^{(d)}_k$. To avoid cluttering the notation, we simply write $\mathbb{E}_q$ in the following; it is clear from the context which variable is excluded. Computing the above equation leads to

$$
\sum_{n} \mathbb{E}_q[[z_{n,d} = k]] \ln \tilde{w}^{(d)}_k - \mathbb{E}_q[\ln \tilde{m}^{(n)}_j] + \text{const}
$$

The term $\mathbb{E}_q[\ln \tilde{w}^{(d)}_k]$ which is due to the normalization of the categorical distribution in Eq. M4 is intractable to compute. In our method, we approximate it via a first-order Taylor expansion, i.e. we use the approximation $\ln(x) \lesssim \ln(x_0) + \frac{x - x_0}{x_0}$. Using this approximation and observing that in Eq. 2 we have to compute the value $-\ln(x)$, we still preserve the lower bound on the log marginal likelihood as required for variational inference. Let $T_{n,d}$ be the point where the Taylor expansion is performed, we get

$$
\mathbb{E}_q[\ln \sum_{j} \tilde{w}^{(d)}_k \cdot \tilde{m}^{(n)}_{m,j}] \leq \\frac{\sum_{j} \mathbb{E}_q[\tilde{w}^{(d)}_k] \cdot \mathbb{E}_q[\tilde{m}^{(n)}_{m,j}] - 1}{T_{n,d}} \quad \text{const}
$$

Note that a different $T_{n,d}$ is used for each object $n$ and dimension $d$. Using this approximation in Equation 2, reordering the terms, and absorbing all terms which are independent of $\tilde{w}^{(d)}_k$ into the constant leads to

$$
\ln q_1^*(\tilde{w}^{(d)}_k) = \left(\mathbb{E}_q[\ln \tilde{m}^{(d)}] - 1 + \sum_n \mathbb{E}_q[[z_{n,d} = k]] \right) \cdot \ln \tilde{w}^{(d)}_k
$$

$$
- \left(1 + \sum_n \mathbb{E}_q[\tilde{m}^{(n)}_j] \cdot \mathbb{E}_q[\tilde{m}^{(n)}_{m,j}] \right) \cdot \tilde{w}^{(d)}_k + \text{const}
$$
Inspecting this result, we see that the optimal distribution of \( q_1^*) \) is a Gamma distribution with shape parameters

\[
\alpha_{d,k}^m = \frac{E_q[\beta_{d,k}^m]}{N} + \sum_n E_q[\delta_{n,d} = k] \quad \beta_{d,k}^m = 1 + \sum_n \frac{E_q[\delta_{n,d}^m]}{\alpha_{n,d,k}^m + \beta_{n,d,k}^m} \quad \text{(U1)}
\]

The equalities \( \ast \) are obtained by normalizing the Dirichlet parameters as given in Equation M5.

**Taylor Expansion Point.** As described above, we perform a Taylor expansion at the point \( T_{n,d} \) (individually for each object \( n \) and dimension \( d \)). Thus, besides the variational parameters \( \Psi \), we now additionally need to optimize the value of \( T_{n,d} \). Analyzing Equation 3 we see that the inequality can be tightened to an equality by selecting:

\[
T_{n,d} = \sum_j E_q[w_j] \cdot E_q[\pi_{j,n,k}] \cdot E_q[m_j] \quad \text{(U2)}
\]

The variables \( T_{n,d} \) are updated after lines 1, 2, and 3 of our processing scheme to keep them up to date.

**Update of \( q_2^* (\pi_{j,n,k}) \).** The update of \( \alpha_{n,k}^m \) and \( \beta_{n,k}^m \) follows the same principle as described above. Please note the change of the term \( \beta_{d,k} \) to \( \sum_d \beta_{d,k} \). Due to the coupling of the Dirichlet parameters as given in Equation M5.

\[
\alpha_{n,k}^m = \frac{1}{N} \sum_d \beta_{d,k} + \sum_n \phi_{n,d,k} \quad \beta_{n,k}^m = 1 + \sum_n \frac{\alpha_{n,k}^m \cdot \beta_{n,k}^m}{\alpha_{n,k}^m - 1} \quad \text{(U3)}
\]

**Update of \( q_3^* (m_k) \).** The above principle cannot be directly applied for the update of \( \alpha_{n,k}^m \) and \( \beta_{n,k}^m \) since the variable \( m_k \) appears in the denominator for the categorical distribution of \( z \). As an alternative, we directly maximize the lower bound given in Equation 1 w.r.t. these variables. The terms of the lower bounding function (Eq. 1) involving \( m_k \) are

\[
E_q[\ln p(m_k | \beta_{d,k}^m)] + E_q[\ln p(\pi, w, m)] - E_q[\ln q_3^*(m_k | \alpha_{n,k}^m, \beta_{n,k}^m)]
\]

There is no closed form solution for the values of \( \alpha_{n,k}^m \) and \( \beta_{n,k}^m \) maximizing this function. Though, fixing \( \alpha_{n,k}^m \) to a value of, e.g., \( a \), the function is concave in \( \beta_{n,k}^m \) and the optimal value for \( \beta_{n,k}^m \) w.r.t. \( a \) can be determined based on the function

\[
b_k(a) = \sqrt{a - 1} \cdot \sqrt{a + (a - 1)c_k^2 + 4 \cdot a \cdot d_k} - (a - 1)c_k
\]

where

\[
c_k = \sum_d \beta_{d,k} - \sum_n \sum_d \phi_{n,d,k} \quad d_k = \sum_n \sum_d \frac{\alpha_{n,k}^m \cdot \beta_{n,d,k}^m}{\alpha_{n,d,k}^m + \beta_{n,d,k}^m} \cdot T_{n,d}
\]

Since \( \beta_{n,k}^m \) is fully determined by the value of \( \alpha_{n,k}^m \), maximizing the lower bounding function is actually only an univariate optimization. After some simplifications, the optimal value of \( \alpha_{n,k}^m \) can be determined by

\[
\alpha_{n,k}^m = \arg \max_{a > 1} \{(c_k - a) \cdot \psi(a) + a + \ln(\Gamma(a)) - \frac{a}{b_k(a)} \cdot \frac{d_k - b_k(a)}{a - 1} - c_k \cdot \ln(b_k(a))\}
\]

where \( \psi \) denotes the digamma function. A local maximum of this function can be found by standard techniques as gradient ascent or brent optimization [11]. The latter one is used in our implementation. Having determined the value for \( \alpha_{n,k}^m \), we can now compute \( \beta_{n,k}^m = b_k(\alpha_{n,k}^m) \).

**Update of \( q_4^* (\beta_{d,k}^m) \).** Since \( q_4 \) corresponds to a Dirac delta function, maximizing the lower bound (Eq. 1) w.r.t. \( \beta_{d,k} \) corresponds to maximizing

\[
E_q[\ln p(m_k^* | \beta_{d,k}^m)] + E_q[\ln p(\pi_k^* | \beta_{d,k}^m)] + E_q[\ln p(\beta_{d,k}^m | \lambda)]
\]

Expanding these terms, one can see that the terms \( \beta_{d,k} \) can be updated for each cluster independently; the dimension-wise values \( \beta_{d,k} \), however, are coupled. For each cluster \( k \) we have to maximize the following function:

\[
G_k(\beta_{d,k}) = \sum_d \left( - \ln \Gamma(\beta_{d,k}) + (\beta_{d,k} - 1) \cdot E_q[\ln w_d] \right) \quad \text{(U6)}
\]

\[
\sum_d \left( - \ln \Gamma(\sum_j \beta_{j,k} / N) + (\sum_j \beta_{j,k} / N) \cdot E_q[\ln \pi_k^*] \right) - \ln \Gamma(\sum_j \beta_{j,k}) + (\sum_j \beta_{j,k} \cdot E_q[\ln m_k]) - \lambda \cdot \sum_d \beta_{d,k}
\]

We use an efficient Newton-Raphson method where the Hessian is inverted in linear time [1] to find a local maximum of the function \( G_k \).

**Update of \( q_5^* (z_{n,d}) \) and \( q_6^* (\Theta_k) \).** The update for the remaining variable depends on the choice of the underlying components’ distributions \( p_k \). In our experiments we focused on Normal distributions where the covariance matrix is a diagonal matrix. Thus, each component \( k \) is parameterized by the means \( \mu_{d,k} \) and variances \( \sigma_{d,k}^2 \) per dimension \( d \). These variables are collected in the parameter vector \( \Theta_k \).

The corresponding conjugate prior for each distribution \( p_k \) is a normal-inverse gamma distribution with prior hyperparameters \( \mu_{d,k}^0, \sigma_{d,k}^0, \alpha_{d,k}^0, \beta_{d,k}^0 \). To realize a non-informative prior we select \( \mu_{d,k}^0, \sigma_{d,k}^0, \beta_{d,k}^0 \rightarrow 0 \) and \( \mu_{d,k}^0 \) is the mean of the data range in dimension \( d \). Since we selected \( q_6 \) to be a Dirac delta function, the corresponding variational parameters are simply denoted as \( \mu_{d,k} \) and \( (\sigma_{d,k})^2 \).

Having fixed this setup, the update of \( q_5^* (z_{n,d}) \) follows the principle of [10] and leads to

\[
\phi_{n,d,k} \propto \exp \left( \psi(\alpha_{n,k}^m) - (\beta_{n,k}^m \cdot \psi(\alpha_{n,k}^m)) - \ln(\beta_{n,k}^m) + \ln(\Gamma(\beta_{n,k}^m) + g_{n,d}) \right)
\]

where

\[
g_{n,d} = - \ln(\sigma_{d,k}^m) - \frac{(X_{n,d} - \mu_{d,k})^2}{2 \cdot (\sigma_{d,k}^m)}
\]

The update of \( \mu_{d,k} \) and \( (\sigma_{d,k})^2 \) can be derived based on the properties of the conjugate prior. Let \( \gamma_{d,k} = \sum_n \phi_{n,d,k} \)
denote the cluster weight and $X_{d,k} = \frac{1}{n} \sum_n \phi_{n,d,k} \cdot X_d^{(n)}$ the (weighted) sample mean (w.l.o.g. $\gamma_{d,k} > 0$; otherwise we simply refer to the hyperparameters). It holds:

$$\hat{\mu}_{d,k} = \frac{\nu_{d,k} \cdot \mu_{d,k} + \gamma_{d,k} \cdot X_{d,k}}{\nu_{d,k} + \gamma_{d,k}} \approx X_{d,k} \quad (U8)$$

$$(\hat{\sigma}_{d,k})^2 \approx \frac{2 \cdot \beta_{d,k}^2}{\gamma_{d,k}} + \frac{1}{\gamma_{d,k}} \sum_n \phi_{n,d,k} \cdot (X_d^{(n)} - X_{d,k})^2 \quad (U9)$$

D. Runtime Complexity

Based on the general processing scheme and the individual update equations, the complexity per iteration can easily be verified as $O(N \cdot D \cdot K)$, i.e. the method scales linear in all parameters. As mentioned, missing data (or sparsity) can easily be incorporated in our model. The update equations can be modified accordingly by just summing over the non-missing values. Consequently, if $M$ denotes the number of non-missing values, the complexity is $O(M \cdot K)$.

IV. Related Work

In this section, we discuss paradigms related to our method for mixed membership subspace clustering.

Subspace clustering: Subspace clustering finds clusters hidden in individual subspace projections of the data. Locally irrelevant attributes, which might cause an obfuscation of the clustering structure in the full-space, are excluded. We refer to the survey [4] for a thorough discussion of subspace clustering methods. The consideration of attribute subsets is highly related to our approach, where we assign a relevance weight to each cluster per dimension. Subspace clustering, in general, allows objects to belong to multiple clusters. The membership information, however, is only binary. Our model allows a more detailed interpretation of cluster memberships by using weight vectors for each object based on the Dirichlet distribution. Finally, while our approach is based on a sound Bayesian model, most subspace clustering approaches do not allow such a probabilistic interpretation. We compare our method with [12] and [13] in the experimental section.

Co-Clustering/Biclustering: Co-clustering is the task of simultaneously clustering the rows and columns of a data matrix. It is highly related to subspace clustering. Differences have been discussed in [4]: While most co-clustering methods treat objects and dimensions (rows and columns) interchangeably, subspace clustering methods treat objects and dimensions differently. Since for many applications the data space and the data objects are indeed different concepts, this distinction leads to a natural interpretation of the clustering result. Accordingly, our proposed method primarily belongs to the paradigm of subspace clustering.

As discussed in [14], co-clustering can roughly be categorized in methods finding patterns with (i) constant values, (ii) constant values on rows or columns, (iii) coherent values, (iv) coherent evolutions. Referring back to the simplified example in Figure 1, our method is mostly related to the second category. In general, however, our model allows to find more complex patterns since the clusters can be represented by arbitrary $D$-dimensional distributions.

Model-based clustering (MBC): This general paradigm assumes the data to be sampled from a statistical model. MBC is very flexible as the modeled distributions can be arbitrarily complex. The detected results allow a sound, probabilistic interpretation.

MBC using ‘traditional’ mixture models: Multivariate mixture models are widely used as a compact representation of the data’s distribution and extensions for high-dimensional data, where clusters are located in subspaces, have been proposed [15]. However, as pointed out in the introduction and in works as [2], [3], these models do not represent mixed membership. The likelihood of an object belonging to a specific cluster corresponds only to the uncertainty inherited from the insufficient information about the ‘true’ clustering structure. The ‘true’ clustering structure in these models is assumed to follow single memberships.

MBC realizing mixed membership: To overcome the above issue, methods as [16], [17], [18], [3] represent mixed membership information. All these approaches propose different ways to model the data; though, sharing a common disadvantage: they do not distinguish between the varying relevance of dimensions for the clusters.

The model [8] considers different weights for the dimensions. It can be seen as the most related competitor to our approach. Several issues, though, make its application for mixed membership subspace clustering questionable: First, unlike to our model where each object is associated with an individual membership vector, the model uses only binary membership information based on a single, global distribution. Second, the method is biased to low-dimensional clusters. As discussed in Sec. II, normalizing the relevance of dimensions per cluster is highly questionable; though, performed by this technique. Third, unlike to our technique where the categorical variable $z_{n,d}$ is derived based on a statistical sound way, the model [8] does not provide such an intuition. Consequently, the weight vectors determined in [8] are hard to interpret and for meaningful knowledge extraction a post-processing needs to be performed. We compare our method against [8] in the experimental section.

MBC for mixed membership co-clustering: The methods [5], [6], [7] perform mixed membership co-clustering. Based on the categorization of [14], they belong to the first class, i.e. they aim at detecting subblocks in the data matrix showing almost constant values. Thus, their mining task is fundamentally different to ours (which, highly simplified, belongs to the second category). To the best of our knowledge there exist no model-based, mixed membership co-clustering method belonging to the second category.

For curiosity we compared our method against [7] in a preliminary study. The quality in all experiments was extremely low. This result is not surprising since the task of [5],
and our clustering task are fundamentally different.

Other principles: The paradigm of multi-view/alternative clustering [19], [20] tackles the challenge of overlapping clusters from a different perspective. The general goal of these methods is to detect multiple partitionings (views) of the data each showing high quality but pairwise dissimilarity. Thus, the results correspond to a very special case of mixed membership since within each partition the clusters are disjoint. Fuzzy clustering allows partial membership of objects to clusters [21]. However, unlike to model-based clustering techniques, the results do not allow a sound, probabilistic interpretation. Mixed membership models are established for document databases (e.g. LDA [1]). The different concepts of data representation (vectors vs. bags of words) permit the direct use of LDA for our task. Furthermore, as mentioned in [5], LDA can be interpreted as a special case of co-clustering of the first category, which, as described above, is inherently different to our task.

V. EXPERIMENTAL EVALUATION

We compare our method, called MMSC, with Power [8], the method being most similar to our approach, and with two subspace clustering approaches: StatPC [12], as a representative for subspace clustering methods allowing mixed membership, and Proclus [13], as a representative for partitioning methods. All approaches were provided with the same number of clusters as input. The additional parameter of Proclus has been tuned to obtain best results. MMSC is run until the variational parameters have been converged (change below 0.01%). As in [8], Power is run for 2000 iterations. All experiments were conducted on 2.6 GHz i5 cores using Java6 64bit. MMSC is parallelized using 2 cores.

Evaluation on synthetic data. We start by evaluating the methods’ performance on synthetic data. Synthetic data is generated based on our generative model. The default data contains 10 clusters, 10 dimensions, and 1000 objects. Since for these data the ground truth is known, clustering quality can be computed. Following the principle of [7], we compute

\[ \frac{1}{n} \cdot \sum_{k,k'} n_{k,k'} \cdot k \cdot k' \cdot \log \left( \frac{1}{k \cdot k'} \right) \]

where \( n_{k,k'} \) denotes the number of object-dimension tuples that belong to cluster \( k \) (of the ground truth) and to cluster \( k' \) (of the detected result), i.e. for each ground truth cluster the best match in the result is determined. This measure accounts for the object groupings as well as the determined subspaces. The normalization ensures that the quality is between 0-1, with 1 being the best result. Since MMSC and Power are probabilistic, we take the expectation over the variables \( z \) to determine which object-dimension pairs belong to which cluster.

In Fig. 4 we increase the size of the database (please note the log-log scale). MMSC’s runtime (top diagram) increases linearly, the slope of the curve is 1. Power shows much higher runtimes. StatPC shows super-linear complexity and it is slower than MMSC for larger datasets. Power and StatPC were not applicable on all data sets due to extreme memory usage (>6GB). The absolute runtime of Proclus is low but increases super-linearly. More important, Proclus obtains only very bad clustering quality (bottom diagram). MMSC shows almost perfect quality. Even though the Power model is highly related to our approach, it is not able to detect the clustering structure.

In Fig. 5 we increase the dimensionality of the data. The results are similar as before: MMSC shows linear increase in runtime, while Power and StatPC increase faster. The clustering quality shows some interesting results. With increasing dimensionality, the results of MMSC get slightly better. The additional dimensions help to detect the underlying clustering structure. In contrast, the quality of the other approaches drops. Particularly, StatPC does not detect any clusters. StatPC’s cluster definition based on hyperrectangles fails to detect clusters in high-dimensional mixed membership data.

In Fig. 6 we increase the number of clusters in the data. Again, MMSC confirms the theoretically derived linear runtime. The runtime results of the other approaches are
similar as before. Since more clusters result in higher overlap between the clusters, the clustering quality of MMSC slightly decreases in this experiment. Though, in any case the quality is higher than the quality of the competing methods. While Power manages to find good results for very few clusters, it cannot handle data with many clusters.

Fig. 7 shows the runtime of MMSC when increasing the number of cores used for parallelization. This experiment was conducted on a Opteron using up to 16 cores. As shown, a huge performance gain is possible. Due to its easy parallelization, MMSC can efficiently be applied on large data.

Besides runtime and quality, we additionally analyzed the number of iterations required for MMSC to converge. For all experiments, the number of iterations showed only small fluctuations with values between 900 to 1400.

Overall, all experiments have shown that MMSC is scalable to large and high-dimensional data sets while simultaneously obtaining high clustering quality.

**Evaluation on real world data.** In the following we describe some interesting results of MMSC on real world data. We start by analyzing the **pendigits data** from the UCI repository that contains handwritten numbers. Since this data is single-label data, i.e. each object is assigned to a single class, we expect to find rather pure membership vectors. To confirm this hypothesis, we computed for each object the entropy of the membership vector determined by MMSC, i.e. $H(\mathbb{E}[\pi^{(n)}])$. Small values indicate, that an object belongs to few clusters only. Fig. 8 shows the results of MMSC where the entropy values are sorted in ascending order. We see that many objects belong to a single cluster only (entropy of zero). The remaining objects also show very low entropy in their membership vectors. The maximal possible entropy, here $\ln(10)$ due to the 10 digits, is never reached. Analyzing the objects that show larger entropy values, one sees that they belong, e.g., to the class of 0 and 8. When writing these numbers, their shape is partly similar leading to a small degree of mixed membership. In general, for this data we do not observe high mixed membership properties.

For the remaining experiments we use multi-label data (http://mulan.sourceforge.net/datasets.html). For these data, each object belongs to multiple classes. We expect to observe a higher degree of mixed membership. In our first experiment, we analyzed the **emotions data**. The data represents music files each described by 72 numerical features. Labels rate the emotions associated to the music. We selected the three labels ‘angry-aggressive’, ‘quiet-still’, ‘amazed-surprised’. The entropy plot of MMSC is shown in Figure 9. In contrast to the previous experiment we observe a much higher entropy, even reaching the maximal possible value. Such objects belong to all clusters equally. This result matches the properties of multi-label data.

**Evaluation on real world data.** In the following we describe some interesting results of MMSC on real world data. We start by analyzing the **pendigits data** from the UCI repository that contains handwritten numbers. Since this data is single-label data, i.e. each object is assigned to a single class, we expect to find rather pure membership vectors. To confirm this hypothesis, we computed for each object the entropy of the membership vector determined by MMSC, i.e. $H(\mathbb{E}[\pi^{(n)}])$. Small values indicate, that an object belongs to few clusters only. Fig. 8 shows the results of MMSC where the entropy values are sorted in ascending order. We see that many objects belong to a single cluster only (entropy of zero). The remaining objects also show very low entropy in their membership vectors. The maximal possible entropy, here $\ln(10)$ due to the 10 digits, is never reached. Analyzing the objects that show larger entropy values, one sees that they belong, e.g., to the class of 0 and 8. When writing these numbers, their shape is partly similar leading to a small degree of mixed membership. In general, for this data we do not observe high mixed membership properties.

For the remaining experiments we use multi-label data (http://mulan.sourceforge.net/datasets.html). For these data, each object belongs to multiple classes. We expect to observe a higher degree of mixed membership. In our first experiment, we analyzed the **emotions data**. The data represents music files each described by 72 numerical features. Labels rate the emotions associated to the music. We selected the three labels ‘angry-aggressive’, ‘quiet-still’, ‘amazed-surprised’. The entropy plot of MMSC is shown in Figure 9. In contrast to the previous experiment we observe a much higher entropy, even reaching the maximal possible value. Such objects belong to all clusters equally. This result matches the properties of multi-label data.

In Figure 11, we plotted the membership vectors $\mathbb{E}[\pi^{(n)}]$ to a simplex, e.g. objects in the corners belong to a single cluster only while objects in the middle belong to each cluster equally. The corners are named with its best matching label. The points are scattered throughout the simplex; though, most objects are located in the corners and/or along the border of the triangle. A very interesting result, is the line between quiet and angry music. We do not observe any points here! Thus, there is no music with emotions associated to angry and quiet but not rated as surprising.

Figure 12 shows the clustering quality of all methods for different multi-label datasets. Since these datasets contain only binary membership information without subspace information, we only evaluate the detected object groupings. To obtain binary membership for MMSC, we binarized the membership vectors such that those clusters are set to 1, which together account for 90% of an object’s member-
Table 12. Clustering quality on multi-label data

<table>
<thead>
<tr>
<th></th>
<th>MMSC</th>
<th>Proclus</th>
<th>StatPC</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emotions (music)</td>
<td>0.72</td>
<td>0.64</td>
<td>0.00</td>
<td>0.60</td>
</tr>
<tr>
<td>Scene (images)</td>
<td>0.76</td>
<td>0.60</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Mediamill (video)</td>
<td>0.50</td>
<td>0.49</td>
<td>0.42</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Figure 13. Varying relevance of dimensions; MMSC on mediamill data

Finally, in Figure 13 we show MMSC’s potential to detect clusters showing varying relevance for the dimensions. For this purpose, we plotted the values $\mathbb{E}[w^{(d)}]$ for each cluster detected in the mediamill data. The data contains videos described by 120 numerical features and labels indicating semantic concepts. We randomly selected 5 concepts and applied MMSC. In the figure, each row is a cluster and each column a dimension. The darker the entry, the more relevant the dimension for the cluster. Obviously, the relevance varies: while some dimensions are important for all clusters (left part of the diagram), other dimensions are used only for few clusters (right part). By exploiting ideas from the subspace clustering paradigm, MMSC is able to detect the varying relevance of dimensions.

Overall, our experiments show that MMSC successfully detects interesting mixed membership subspace clusters on a variety of real world data sets.

VI. CONCLUSION

We proposed the MMSC approach that successfully exploits the model-based clustering paradigm for the detection of mixed membership clusters in vector data. Our Bayesian framework accounts for the observation that objects might belong to multiple clusters with different degrees and that the relevance of dimensions might vary for each cluster individually. For an efficient learning of our model, we used the principle of variational inference and we derived the required update equations. In our empirical study on synthetic and real-world data, we demonstrated the strengths of MMSC in comparison to related techniques.

Acknowledgments. Stephan Günnemann has been supported by a fellowship within the postdoc-program of the German Academic Exchange Service (DAAD).

REFERENCES