SMVC: Semi-Supervised Multi-View Clustering in Subspace Projections

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

Ines Färber Matthias Rüdiger Thomas Seidl
RWTH Aachen University, Germany
{faerber, ruediger, seidl}@cs.rwth-aachen.de

ABSTRACT

Since data is often multi-faceted in its very nature, it might not adequately be summarized by just a single clustering. To better capture the data's complexity, methods aiming at the detection of *multiple, alternative clusterings* have been proposed. Independent of this research area, semi-supervised clustering techniques have shown to substantially improve clustering results for *single-view clustering* by integrating prior knowledge. In this paper, we join both research areas and present a solution for integrating prior knowledge in the process of detecting multiple clusterings.

We propose a Bayesian framework modeling multiple clusterings of the data by multiple mixture distributions, each responsible for an individual set of relevant dimensions. In addition, our model is able to handle prior knowledge in the form of instance-level constraints indicating which objects should or should not be grouped together. Since a priori the assignment of constraints to specific views is not necessarily known, our technique automatically determines their membership. For efficient learning, we propose the algorithm SMVC using variational Bayesian methods. With experiments on various real-world data, we demonstrate SMVC's potential to detect multiple clustering views and its capability to improve the result by exploiting prior knowledge.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications —*Data mining*; I.2.6 [**Artificial Intelligence**]: Learning

Keywords

semi-supervised learning; subspace clustering; constraints

1. INTRODUCTION

Clustering aims at grouping data instances based on their similarity. For complex data, however, the similarity often depends on the point of view. In a customer database, for example, users might be grouped according to their demographic profile or according to their buying patterns. In a document database, different groups might reflect the documents' subjects or their writing style. Thus, depending on the application and the user's preferences, a single grouping does not capture all aspects but *multiple, alternative clustering solutions* are required. The emerging research field of multi-view or alternative clustering [27] addresses this challenge by finding multiple high quality clusterings.

On the other hand, semi-supervised clustering techniques [8] try to incorporate the user's preferences by exploiting prior knowledge during the clustering process. For traditional single-view clustering, these techniques have shown to substantially increase the clustering results. Motivated by the success of both research areas, we propose a semi-supervised multi-view clustering technique. Our goal is to exploit user provided prior knowledge to enhance the results of multiple, alternative clusterings.

For semi-supervised clustering, it is crucial that the user can provide supervision in an easy and understandable way. While cluster level constraints, such as the clusters' sizes, positions, or distributions, usually require an abstract understanding of the desired clustering structure, instance level constraints which, e.g., indicate partial information about cluster memberships, are much more intuitive. A popular way of modeling such prior information is via equivalence constraints, which indicate for pairs of instances whether they should belong to the same cluster (must-link constraint) or to different clusters (cannot-link). Even though lacking a full understanding of the clustering structure, this allows the user to partly specify her intuition by indicating for selected object pairs their pairwise cluster relation. Since in many cases these user constraints express a belief rather than certainty, we use the concept of soft constraints, where mistakes are possible and a complete fulfillment is not enforced.

The transfer of the semi-supervised clustering principle to the multi-view case poses a severe challenge, particularly regarding the multi-faceted nature of the data. One user might for example see the similarity of two movies based on their cast, while another user might foreground their dissimilarity based on differing genres. It, therefore, might remain unclear to which view specific constraints refer to. In particular, when naively assigning all constraints to a single view, a large proportion of the constraints might be conflicting such that even a relaxation to soft constraints will not be sufficient anymore. Therefore, the challenge in semi-supervised multi-view clustering is not only to optimize the clustering such that constraints are optimally fulfilled but also to *learn the affiliation of constraints to views.*

It has to be highlighted that some of the *sequentially* working multi-view clustering approaches (which iteratively find

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623734.

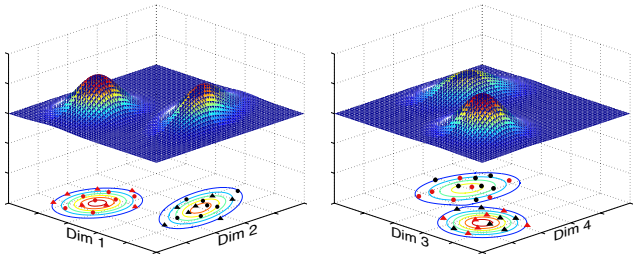


Fig. 1: Example for the multi-view scenario

one clustering at a time) (e.g. [4, 30]) already work based on instance level constraints to incorporate the feedback of rejected prior clusterings via cannot-link constraints. These constraints, however, are used for a different goal: they guide the clustering method to find *a single* new clustering. Thus, all constraints need to refer to this single clustering, and none of the previous clusterings can be affected by these constraints. In contrast, our aim is to incorporate instance level constraints which might improve the *overall* result of all clusterings. It becomes apparent, that in this case we have to rely on a multi-view clustering technique which detects all clusterings *simultaneously*.

Only few approaches for simultaneous multi-view clustering have been proposed (e.g. [28, 23, 22, 21]). Here, the inevitable connection of multi-view clustering and subspace clustering has been observed first [28, 22, 21], which later also influenced sequentially working approaches like [15]. Subspace clustering assumes each cluster to have an individual set of relevant data attributes, which corresponds well with the motivation of multi-view clustering that different views on the data (i.e. considering different characteristics of the data) might reveal different clustering structures.

In this work we join the three paradigms of simultaneous multi-view clustering, subspace clustering, and constraint-based clustering. We present a Bayesian framework that models the different clustering views via several multivariate mixture distributions located in subspace projections (cf. Figure 1). Each object follows multiple components, each in a different mixture model, each defining a distribution only for a certain view (i.e. subspace) of the data, and each representing a different role of the object. We integrate the optimal fulfillment of user provided instance level constraints into the Bayesian learning process, where we tackle the challenge of automatically learning the responsibility of views for specific constraints. Our contributions are:

- *Multiple clusterings:* We propose a sound Bayesian model which represents multiple clusterings via individual mixture models, each representing a distinct view.
- *Semi-supervision:* Our model incorporates prior knowledge in form of (soft) must-link and cannot-link instance level constraints. Our method automatically learns the assignment of these constraints to specific views if their responsibility is not explicitly specified.
- *Algorithm design:* We present an efficient algorithm based on the principle of variational inference for learning our model.
- *Effectiveness:* We analyze the effectiveness of our method on various datasets and show its potential to increase the clustering result by using prior knowledge.

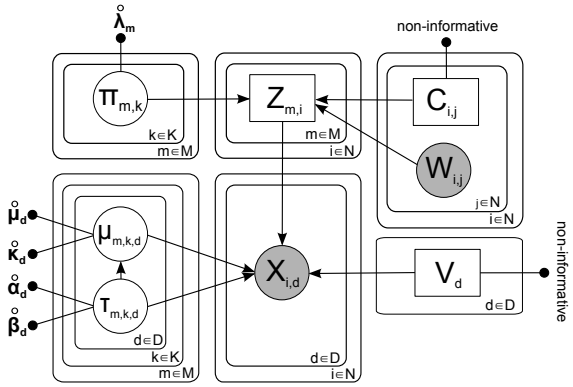


Fig. 2: Graphical model of our method. Rectangles denote discrete random variables, circles continuous random variables, and black dots (deterministic) hyperparameters of the prior distributions.

2. BAYESIAN FRAMEWORK

In this section, we introduce a Bayesian framework for semi-supervised multi-view clustering. An overview of our framework is given by the graphical model depicted in Fig. 2. While this section introduces the generative process of our model, we describe in Section 3 how to learn the model's parameters given a set of observations. Following convention, we do not distinguish between a random variable x and its realization $x = \mathbf{x}$ 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 random variables $\{y_{i,d}\}$ with i in the corresponding index domain, and Y is an abbreviation for the set $y_{*,*}$.

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

Multiple Mixture Models. The general idea of our method is to represent the multiple clusterings of the data by multiple mixture models, each located in a different subspace projection (cf. Figure 1). In this work, we focus on Gaussian mixture models; extensions to other distributions are straightforward. Following standard principles, each of the M mixture models is based on K components, where each of these components is associated with a mean and a covariance/precision matrix. To reduce the number of parameters to be estimated, we focus on diagonal precision matrices. Thus, for a Bayesian treatment, we introduce the random variables

$$(\mu_{m,k,d}, \tau_{m,k,d}) \sim \mathcal{NG}(\mathring{\mu}_d, \mathring{\kappa}_d, \mathring{\alpha}_d, \mathring{\beta}_d) \tag{1}$$

where $\mu_{m,k,d}$ is the mean of component k in dimensions d for clustering m, and $\tau_{m,k,d}$ the corresponding precision. We select the normal-gamma distribution $\mathcal{NG}$ as a prior since it represents the variables' conjugate prior. The hyperparameters denoted by $\mathring{*}$ can be used to control the mixture models' components if some prior knowledge is available. Per default, we choose least informative priors by selecting $\mathring{\kappa}_d, \mathring{\alpha}_d \to 0$ and setting $\mathring{\mu}_d / \mathring{\beta}_d$ to be the sample mean/sum of squared deviations in dimension d.

Besides the components parameters, each mixture model is associated with a corresponding random variable representing the mixture weights. Obviously, since we want to

find multiple different clusterings, these weights can be different for each view. We use the random variable

$$\vec{\pi}_m \sim Dir(\mathring{\lambda}) \tag{2}$$

where $\pi_{m,k}$ is the weight of component k in clustering m. Due to conjugate properties, we use a Dirichlet distribution as its prior. Again, in our study, we use a non-informative prior by selecting $\mathring{\lambda} = 1$ since a priori no knowledge about the cluster sizes is given.

Integrating Subspaces. To detect the data's multiple views, we refer to the principle of subspace clustering. Our goal is to assign each mixture model to a specific subspace projection, which it describes well. Since the relevant dimensions of the mixtures are a priori not known, we learn them with our method. Therefore, we introduce the random variable

$$v_d \sim Categorical(\vec{r}_d) \tag{3}$$

to indicate which of the M clusterings is responsible for a specific dimension d. The vector $\vec{r}_d \in [0 \dots 1]^M$ (with $\sum r_{k,m} = 1$) can be used to give some prior knowledge which dimension belongs to which view. Again, we use a constant non-informative prior, i.e. $r_{k,m} = 1/M$.

Knowing about the subspaces as well as the mixture models' parameters, we are now able to generate observations which show multiple clustering structures: We denote with $z_{m,i}$ the random variable indicating to which cluster an object i belongs to in clustering m, i.e.

$$z_{m,i} \sim Categorical(\vec{\pi}_m) \tag{4}$$

Note that for each view m, the object might follow a different cluster, i.e. $z_{m,i} \neq z_{m',i}$ is possible. Thus, in each view the object might be grouped together with different objects. This idea is illustrated in Figure 1: the grouping on the left differs from the one on the right. Given $z_{m,i}$, the attribute value of object i in dimension d is drawn according to

$$x_{i,d} \sim \mathcal{N}(\mu_{m,k,d}, \tau^{-1}_{m,k,d}) \text{ with } m = v_d \text{ and } k = z_{m,i} \tag{5}$$

That is, we use the clustering m which is responsible for dimensions d and the corresponding component k the object belongs to in this view.

Integrating User Constraints. So far, our model corresponds to a completely unsupervised technique for finding multiple clusterings. As a major advancement, we now integrate user provided prior-knowledge. As discussed, we aim to support the concept of instance level constraints. More precisely, we support the idea of soft constraints between pairs of objects that indicate whether the objects should or should not be grouped together. We selected this type of semi-supervision since it reflects an intuitive understanding of clustering and is easy to specify for the user.

The user can provide a constraint between the objects i and j via a weight $w_{i,j}$. If the weight is positive, the user indicates that there should exist a clustering where the objects are grouped together. If the weight is negative, the user indicates that there should exist a clustering where i and j are not grouped together. Different magnitudes of the weights can be used to indicate the different importance or relevance of the constraints.

At this point it is crucial to keep in mind that we are interested in finding multiple, alternative clusterings: A constraint between i and j means that *there exists a view* where the constraint is fulfilled. We do not require that i and j are grouped together in *all views*, which actually would contradict the fundamental assumption for multi-view scenarios that clusterings of different views differ and contain alternative knowledge. Forcing constraints to be valid for all views would be too restrictive. Furthermore, we argue that the user is generally not aware of the details of all possible groupings. Thus, the user should not define constraints restricting views that he does not understand. Accordingly, for each constraint, we are interested in finding (at least) one clustering fulfilling this constraint.

Resulting from this principle, another challenge of our method becomes apparent: we have to determine the clustering which is responsible for a specific constraint. In the following, we show how to model all these aspects.

As mentioned, the constraints are modeled via weights. In our model, we represent them via a symmetric matrix W of size $N \times N$, where entries with weight zero indicate no prior knowledge about the corresponding pairs of objects. In practice, we can use a *sparse representation* of the matrix which only encodes the given constraints and allows for an efficient processing. Interesting to note is that the (observed) matrix W appears in our grapical model as one of the root nodes (cf. Figure 2), and not as a leaf like X. As shown, the weights influence the grouping Z of the objects.

Additionally, we introduce the categorical random variables $c_{i,j}$ (due to the symmetry of the weights, we only need to consider $i < j$). These variables indicate which view is responsible for a specific constraint. That is, we have

$$c_{i,j} \sim Categorical(\vec{h}^{(i,j)}) \tag{6}$$

where $\vec{h}^{(i,j)} \in [0...1]^M$ with $\sum_{m \in M} h_m^{(i,j)} = 1$. The user can use $\vec{h}^{(i,j)}$ to express some further prior knowledge about the constraint between object i and j. If the user, for example, knows that a set of constraints should most likely belong to one view, the h vectors can be selected accordingly. Per default, we assume that no knowledge about the assignment of constraints to views is known, i.e. we use $h_m^{(i,j)} = 1/M$.

Given W and C, how can we use their values to influence the clustering structure of the data? Our idea is to add a bias to the probability distribution of the $z_{m,j}$. The probability of generating a clustering that matches the constraints should be higher than the probability of a clustering which violates the constraints. Particularly, this results in a dependency between the variables $z_{m,*}$ which is guided by the constraints. We define

$$p(z_{m,*} \mid \vec{\pi}_m, W, C) \propto \prod_{i=1}^{N} \pi_{m,z_{m,i}} \cdot \prod_{i=1}^{N} \prod_{\substack{j>i \\ c_{i,j}=m}}^{N} e^{w_{i,j} \cdot \delta(z_{m,i}, z_{m,j})} \tag{7}$$

Here, $\delta(z_{m,i}, z_{m,j})$ denotes the Kronecker delta, which evaluates to 1 if both objects are located in the same cluster (in view m), and 0 otherwise. Please note that Equation 7 is the joint distribution for all $z_{m,*}$.

The first part of the equation corresponds to the mixture weights as used in standard mixture models. If all $w_{i,j} = 0$, Equation 4 and 7 are equivalent. The second part models the bias to specific groupings: As one can see, if $w_{i,j}$ is positive and the objects are located in the same cluster, the probability of selecting this grouping increases. Accordingly, if $w_{i,j}$ is negative, one would decrease the probability of clusterings where i and j are grouped together. A similar principle was used in [25, 7] for single-view clustering.

Important to mention is that the second part of the equation incorporates the automatic assignment of constraints to views. The constraint between i and j adds a bias to the clustering structure in view $c_{i,j} = m$ only. In accordance to our discussion above, the other views are not affected.

Given the new definition for the distribution of Z, the actual observations are, as before, generated according to Equation 5. Overall, our model combines the principle of multiple clusterings in subspace projections with the paradigm of semi-supervised clustering and automatically assigns constraints to their responsible views.

3. THE SMVC ALGORITHM

While the previous section has focused on the model's generative process, we now present our learning technique. That is, *given* a set of observations X and a set of constraints W, we infer the model's parameters. Our method is called SMVC (Semi-Supervised Multi-View Clustering).

3.1 Variational Inference

The general inference problem we have to solve is to determine the distribution $p(Y|X,W)$, where $Y=\{V,Z,C,\vec{\pi},\mu,\tau\}$ 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. Since computing $p(Y|X,W)$ is intractable, we compute an approximation based on the principle of variational inference [10]: we approximate $p(Y|X,W)$ by a tractable family of parametrized distributions $q(Y|\Psi)$. The parameters Ψ 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 Ψ. Using Jensen's inequality, minimizing the KL divergence is equivalent to maximizing the following lower bound on the log marginal likelihood [10]:

$$\mathcal{L}(X,W;\Psi) = \mathbb{E}_q[\ln p(X,W,Y)] - \mathbb{E}_q[\ln q(Y|\Psi)] \quad (8)$$

where $\mathbb{E}_q[.]$ denotes the expectation w.r.t. the q distribution.

Following primarily the idea of mean field approximation, we assume the function q to factorize in

$$p(Y \mid X,W) \approx q(Y|\Psi) := \prod_d q_1(v_d) \cdot \prod_m \prod_i q_2(z_{m,i})$$
$$\cdot \prod_i \prod_{j>i} q_3(c_{i,j}) \cdot \prod_m q_4(\vec{\pi}_m) \cdot \prod_m \prod_k \prod_d q_5(\mu_{m,k,d}, \tau_{m,k,d})$$

As we will later see, assuming the above factorization, the optimal variational distributions have the form

$$\begin{aligned} q_1(v_d) &= Categorical(v_d \mid \phi_{d,1}, ..., \phi_{d,M}) \\ q_2(z_{m,i}) &= Categorical(z_{m,i} \mid \psi_{m,i,1}, ..., \psi_{m,i,K}) \\ q_3(c_{i,j}) &= Categorical(c_{i,j} \mid \xi_{i,j,1}, ..., \xi_{i,j,M}) \\ q_4(\vec{\pi}_m) &= Dir(z_{m,i} \mid \vec{\lambda}_m) \\ q_5(\mu_{m,k,d}, \tau_{m,k,d}) &= \mathcal{NG}(\mu_{m,k,d}, \tau_{m,k,d} \mid \\ & \qquad \tilde{\mu}_{m,k,d}, \tilde{\kappa}_{m,k,d}, \tilde{\alpha}_{m,k,d}, \tilde{\beta}_{m,k,d}) \end{aligned}$$

where $\Psi = \{\phi, \psi, \xi, \vec{\lambda}, \tilde{\mu}, \tilde{\kappa}, \tilde{\alpha}, \tilde{\beta}\}$ are the variational parameters to be optimized. Note that each distribution has its own variational parameters [10]. Thus, e.g. the functions $q_1(v_d)$ and $q_1(v_{d'})$, are not necessarily identical. This extra degree of freedom allows to find a good approximation between q and p. As discussed in Section 2, for $c_{i,j}$, i.e. the function q_3, we only need to consider pairs i,j with $w_{i,j} \neq 0$.

General Processing Scheme. We use an iterative coordinate ascent method to maximize Equation 8 w.r.t. the parameters Ψ (the update equations follow in Section 3.2). The processing scheme is as follows:

while *not converged* **do**
- **for** $i,j \in N : j > i \wedge w_{i,j} \neq 0$ **do** update $\xi_{i,j,*}$ — Eq. 10
- **for** $d \in D$ **do** update $\phi_{d,*}$ — Eq. 11
- **for** $m \in M, i \in N$ **do** update $\psi_{m,i,*}$ — Eq. 12
- **for** $i \in N, m \in M$ **do** update $\vec{\lambda}_m$ — Eq. 13
- **for** $m \in M, k \in K, d \in D$ **do** update $\tilde{\mu}_{m,k,d}, \tilde{\kappa}_{m,k,d}, \tilde{\alpha}_{m,k,d}, \tilde{\beta}_{m,k,d}$ — Eq. 14

Note that due to the properties of variational inference [10], it is guaranteed that the method converges. In practice, we assume convergence if the change in the lower bound on the marginal likelihood is below than 0.01. Additionally, to avoid the problem of local minima, we enhance the processing scheme by gradually increasing the importance of the constraints. That is, starting with low weights, we linearly increase the values $w_{i,j}$ until they reach the user specified scores. For initializing our method, we exploit the same principle as described in [21]. The random variable C/q_3 is initialized randomly based on its prior distribution.

3.2 Update Equations

We briefly present the update equations required for the coordinate ascent method. We primarily follow the principle of [10]: The optimal distribution for $q_x(B)$ can be determined by

$$\ln q_x^*(B) = \mathbb{E}_{q \setminus B}[\ln p(X,Y,W)] + \mathcal{C} \quad (9)$$

Here, the constant $\mathcal{C}$ absorbs all terms which are independent of B and, thus, do not affected the optimal distribution of q_x. $\mathbb{E}_{q \setminus B}[.]$ denotes the expectation w.r.t. the distribution q taken over all variables Y except of B. To avoid cluttering the notation, we simply write $\mathbb{E}_q$ in the following (it is clear from the context which variable is excluded).

Updating the constraint responsibility. Let $[\![.]\!]$ denote the Iverson bracket. We can rewrite Equation 7 as follows

$$\prod_{i=1}^{N} \prod_{k=1}^{K} \pi_{m,k}^{[\![z_{m,i}=k]\!]} \cdot \prod_{i=1}^{N} \prod_{j>i}^{N} \prod_{k=1}^{K} e^{w_{i,j}[\![z_{m,i}=k]\!][\![z_{m,j}=k]\!][\![c_{i,j}=m]\!]}$$

This formulation makes it easier to derive the following results. Accordingly, we can rewrite the remaining equations.

The optimal distribution for $q_3(c_{a,b})$ (with $a < b$) can be obtained via Eq. 9. Removing all terms which are independent of $c_{a,b}$ and using the above reformulation, we get

$$\begin{aligned} & \log q_3^*(c_{a,b} = y) \\ &= \mathbb{E}_q[\log (P(c_{i,j})P(Z|\pi,C,W))] + \mathcal{C} \\ &= \mathbb{E}_q[\log \frac{1}{M}] + \mathbb{E}_q[\log \prod_{m=1}^{M} \Big(\prod_{i=1}^{N} \prod_{k=1}^{K} \pi_{m,k}^{[\![z_{m,i}=k]\!]} \cdot \prod_{i=1}^{N} \prod_{j>i}^{N} \\ &\qquad \cdot \prod_{k=1}^{K} e^{w_{i,j}[\![z_{m,i}=k]\!][\![z_{m,j}=k]\!][\![c_{i,j}=m]\!]} \Big)] + \mathcal{C} \\ &= \mathbb{E}_q[\sum_{m=1}^{M} \sum_{k=1}^{K} \log e^{w_{a,b}[\![z_{m,a}=k]\!][\![z_{m,b}=k]\!][\![c_{a,b}=m]\!]}] + \mathcal{C} \\ &= w_{a,b} \sum_{k=1}^{K} \mathbb{E}_q[\![z_{y,a} = k]\!] \cdot \mathbb{E}_q[\![z_{y,b} = k]\!] + \mathcal{C} \end{aligned}$$

Since $c_{a,b}$ has a finite domain, the distribution q_3 is a categorical distribution. Renaming the variables, the optimal hyperparameters of the distribution $q_3(c_{i,j})$ are given by

$$\xi_{i,j,m} \propto exp^{\left(w_{i,j} \sum_{k=1}^{K} \mathbb{E}_q[\![z_{m,i}=k]\!] \cdot \mathbb{E}_q[\![z_{m,j}=k]\!]\right)} \quad (10)$$

where $\sum_m \xi_{i,j,m} = 1$. The occurring expectations can be replaced by the known expectations of the variational distributions (cf. appendix). Intuitively, the parameter $\xi_{i,j,m}$ shows the probability of assigning the constraint between i and j to the view m.

Updating the views. Computing Eq. 9 for $q_1(v_d)$ and removing all terms which are independent of v_d leads to

$$\begin{aligned}
&\ln q_1^*(v_d = y) \\
&= \mathbb{E}_q[\log\left(P(x_{*,d}|v_d, Z, \mu, \tau)P(v_d)\right)] + \mathcal{C} \\
&= \mathbb{E}_q[\log \prod_{m=1}^{M}\prod_{i=1}^{N}\prod_{k=1}^{K} \mathcal{N}(x_{i,d}|\mu_{m,k,d}, \tau_{m,k,d}^{-1})^{[\![v_d=m]\!][\![z_{m,i}=k]\!]}] \\
&\qquad + \mathbb{E}_q[\log \frac{1}{M}] + \mathcal{C} \\
&= \mathbb{E}_q[\log \prod_{i=1}^{N}\prod_{k=1}^{K} \mathcal{N}(x_{i,d}|\mu_{y,k,d}, \tau_{y,k,d}^{-1})^{[\![z_{y,i}=k]\!]}] + \mathcal{C} \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} \mathbb{E}_q[\![z_{y,i} = k]\!] \cdot f(y,k,d,i) + \mathcal{C}
\end{aligned}$$

Here, we used the definition

$$\begin{aligned}
&f(m,k,d,i) := \mathbb{E}_q[\mathcal{N}(x_{i,d}|\mu_{m,k,d}, \tau_{m,k,d}^{-1})] \\
=&\mathbb{E}_q[\log \sqrt{\frac{\tau_{y,k,d}}{2\pi}} e^{\frac{-(x_{i,d}-\mu_{y,k,d})^2\tau_{y,k,d}}{2}}] \\
=&\frac{1}{2}\mathbb{E}_q[\log \frac{\tau_{y,k,d}}{2\pi}] + \frac{1}{2}\mathbb{E}_q[-(x_{i,d}-\mu_{y,k,d})^2\tau_{y,k,d}] \\
=&\frac{1}{2}\cdot(\mathbb{E}_q[\log\tau_{y,k,d}] - x_{i,d}^2\cdot\mathbb{E}_q[\tau_{y,k,d}] + 2\cdot x_{i,d}\cdot\mathbb{E}_q[\mu_{y,k,d}\cdot\tau_{y,k,d}] \\
&\qquad - \mathbb{E}_q[\mu_{y,k,d}^2\cdot\tau_{y,k,d}] - \mathbb{E}_q[\log 2\pi])
\end{aligned}$$

Thus, q_1 is a categorical distribution and the optimal hyperparameters for $q_1(v_d)$ are given by

$$\phi_{d,m} \propto exp^{\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}_q[\![z_{m,i}=k]\!]\cdot f(m,k,d,i)} \tag{11}$$

where $\sum_m \phi_{d,m} = 1$.

Updating the cluster indicator. The same principle can be applied for the cluster indicator variable. We obtain:

$$\begin{aligned}
&\log q_2^*(z_{m,a} = y) \\
&= \mathbb{E}_q[\log\left(P(x_{a,*}|V, Z, \mu, \tau)P(Z|\pi, C, W)\right)] + \mathcal{C} \\
&= \mathbb{E}_q[\log \prod_{d=1}^{D}\prod_{k=1}^{K} \mathcal{N}(x_{a,d}|\mu_{m,k,d}, \tau_{m,k,d}^{-1})^{[\![v_d=m]\!][\![z_{m,a}=k]\!]}] + \\
&\quad \mathbb{E}_q[\log \prod_{k=1}^{K}\prod_{i=1}^{N} \pi_{m,k}^{[\![z_{m,i}=k]\!]} \prod_{i=1}^{N}\prod_{j>i}^{N} e^{w_{i,j}[\![z_{m,i}=k]\!][\![z_{m,j}=k]\!][\![c_{i,j}=m]\!]}] \\
&= \sum_{d=1}^{D} \mathbb{E}_q[\![v_d = m]\!]\, \mathbb{E}_q[\log \mathcal{N}(x_{a,d}|\mu_{m,y,d}, \tau_{m,y,d}^{-1})] + \mathbb{E}_q[\log \pi_{m,y}] \\
&\qquad + \sum_{i=1}^{N}\sum_{j>i}^{N} w_{i,j}\, \mathbb{E}_q[\![z_{m,i} = y]\!]\, \mathbb{E}_q[\![z_{m,j} = y]\!]\, \mathbb{E}_q[\![c_{i,j} = m]\!] + \mathcal{C} \\
&= \sum_{d=1}^{D} \mathbb{E}_q[\![v_d = m]\!] \cdot f(m,y,d,a) + \\
&\qquad \mathbb{E}_q[\log \pi_{m,y}] + \sum_{j\neq a}^{N} w_{a,j}\, \mathbb{E}_q[\![z_{m,j} = y]\!]\, \mathbb{E}_q[\![c_{a,j} = m]\!] + \mathcal{C}
\end{aligned}$$

Here, we exploit the symmetry of $w_{i,j}$ and the definition of f as given above. Note again, that we do not actually need to sum over all $j \neq a$ when using a sparse encoding of the matrix W. It is sufficient to iterate over those j for which a constraint with a is given. Similar as before, the optimal hyperparameters for $q_2(z_{m,i})$ are given by

$$\begin{aligned}
\psi_{m,i,k} \propto exp\Big(&\sum_{d=1}^{D} \mathbb{E}_q[\![v_d = m]\!] \cdot f(m,k,d,i) \\
&+ \mathbb{E}_q[\log \pi_{m,k}] + \sum_{j\neq i}^{N} w_{i,j}\, \mathbb{E}_q[\![z_{m,j} = k]\!]\, \mathbb{E}_q[\![c_{i,j} = m]\!]\Big)
\end{aligned} \tag{12}$$

with $\sum_k \psi_{m,i,k} = 1$.

Updating the mixing weights. The mixing weights are continuous. Since we selected a conjugate prior in our model, it follows:

$$\begin{aligned}
&\log q_4^*(\vec{\pi}_m) \\
&= \mathbb{E}_q[\log\left(P(\pi_m)P(z_{m,*}|\pi, C, W)\right)] + \mathcal{C} \\
&= \mathbb{E}_q[\log\left(\frac{\Gamma(\mathring{\lambda}K)}{\Gamma(\mathring{\lambda})^K}\prod_{k=1}^{K}\pi_{m,k}^{\mathring{\lambda}-1}\right)] + \mathbb{E}_q[\log\Big(\prod_{i=1}^{N}\prod_{k=1}^{K}\pi_{m,k}^{[\![z_{m,i}=k]\!]}\prod_{i=1}^{N} \\
&\qquad \cdot \prod_{j>i}^{N}\prod_{k=1}^{K} e^{w_{i,j}[\![z_{m,i}=k]\!][\![z_{m,j}=k]\!][\![c_{i,j}=m]\!]}\Big)] + \mathcal{C} \\
&= \sum_{k=1}^{K}(\mathring{\lambda}-1)\mathbb{E}_q[\log\pi_{m,k}] + \sum_{i=1}^{N}\sum_{k=1}^{K}\mathbb{E}_q[\![z_{m,i}=k]\!]\mathbb{E}_q[\log\pi_{m,k}] + \mathcal{C} \\
&= \sum_{k=1}^{K}\left((\mathring{\lambda}-1) + \sum_{i=1}^{N}\mathbb{E}_q[\![z_{m,i}=k]\!]\right)\cdot\mathbb{E}_q[\log\pi_{m,k}] + \mathcal{C}
\end{aligned}$$

As seen, the optimal distribution for q_4 is a Dirichlet distribution, where the hyperparameters are given by

$$\tilde{\lambda}_m[k] = \mathring{\lambda} + \sum_{i=1}^{N}\mathbb{E}_q[\![z_{m,i}=k]\!] \tag{13}$$

Updating the mixture components. Updating the mean and precision of each mixture component follows the standard principle of variational inference in a conjugate setting. Let $u_{m,k} = \sum_{i=1}^{N}\mathbb{E}_q[\![z_{m,i}=k]\!]$ be the unnormalized weight of a cluster and $\bar{x}_{m,k,d} = \frac{1}{u_{m,k}}\sum_{i=1}^{N} x_{i,d}\mathbb{E}_q[\![z_{m,i}=k]\!]$ its weighted mean in dimension d (when considering the expectation w.r.t. q). Using conjugacy, it follows that the optimal hyperparameters of the distribution q_5 are given by

$$\begin{aligned}
\tilde{\mu}_{m,k,d} &= \frac{\mathring{\kappa}_d\ \mathring{\mu}_d + u_{m,k}\ \bar{x}_{m,k,d}}{\mathring{\kappa}_d + u_{m,k}} \qquad \tilde{\kappa}_{m,k,d} = \mathring{\kappa}_d + u_{m,k} \\
\tilde{\alpha}_{m,k,d} &= \mathring{\alpha}_d + \frac{u_{m,k}}{2}
\end{aligned} \tag{14}$$

$$\tilde{\beta}_{m,k,d} = \mathring{\beta}_d + \frac{1}{2}\sum_{i=1}^{N}(x_{i,d}-\bar{x}_{m,k,d})^2 + \frac{\mathring{\kappa}_d\ u_{m,k}}{\mathring{\kappa}_d + u_{m,k}}\frac{(\bar{x}_{m,k,d}-\mathring{\mu}_d)^2}{2}$$

3.3 Complexity and Summary

Inspecting the individual update equations, it becomes apparent that each iteration of our algorithm runs in time $\mathcal{O}(M \cdot N \cdot K \cdot (D + W))$, where W denotes the number of constraints. Thus, we obtain a *linear complexity* in all important parameters.

Overall, our method efficiently computes an approximation of the posterior distribution $p(Y|X, W)$ which shows us the multiple clustering structures, their relevant subspaces, and the assignment of constraints to views.

4. RELATED WORK

Our approach is related to four main paradigms in the field of cluster analysis, as we will discuss in the following. Table 1 shows on overview of the related works and their corresponding properties.

Subspace clustering. For traditional full-space clustering, a large proportion of irrelevant attributes can cause an obfuscation of the clustering structure. The underlying assumption of subspace clustering (co-clustering/bi-clustering) [24, 2, 26] is that the set of irrelevant attributes might differ for each cluster. These locally irrelevant attributes hinder a meaningful global dimensionality reduction [29] and make traditional, full-space approaches futile. The consideration of attribute subsets is highly related to our multi-view scenario, since different views of the data are most likely attributable to different characteristics. However, subspace clustering does not realize a grouping of clusters to represent alternative views as required for multi-view clustering.

Multi-view clustering. The paradigm of multi-view or alternative clustering can be categorized in three types [27]: Approaches of the first category, e.g. [23, 4, 13, 14], operate in the full-space and, therefore, suffer from similar problems as traditional clustering. Furthermore, they usually aim at finding just two alternative clusterings. The second category's representatives iteratively determine an alternative clustering based on the previous one via space transformations such as PCA [11, 15] or distortion of the distance function [30]. They do not globally/simultaneously optimize the whole set of all clusterings. Since previous clustering solutions serve as guidance for the discovery of new clustering structures, these approaches can partially be categorized as semi-supervised. However, the constraints affect only the solution of the single, next clustering and, thus, already detected solutions cannot benefit from them. Additionally, distortions of the original space usually hinder an intuitive interpretation of the clustering result. Contrarily, axis-parallel projections of the data as used in our approach allow an easy interpretation. The third category, which is mostly related to our approach, represents methods that *simultaneously* reveal all clusterings by analyzing axis-parallel subspace projections [28, 22, 21]. These approaches do not incorporate any user knowledge. With our SMVC approach, we want to examine the usefulness of instance level constraints for the process of simultaneous multi-view clustering.

Model-based clustering. This general paradigm assumes the considered data to be sampled from a statistical model. Several approaches for estimating the parameters of the underlying probability distributions, e.g., to maximize the log-likelihood of the data, were proposed including the EM or variational inference [10]. Model-based clustering is very flexible as the modeled distributions can be arbitrarily complex. Traditionally, such approaches use a *single* mixture distribution (which spans across all dimensions of the data space). Even though each observation might be associated with a membership degree (e.g. the likelihood of belonging to a cluster), this principle does not capture the idea of generating objects through multiple components as required for the multi-view scenario. To overcome this issue, a few models [17, 5, 19] try to represent such multi-component membership (i.e. overlapping clusters). Although, these models lead to results where an object might take multiple roles within a single view, they do not account for the principle of multiple views. So far, MVGen [21] is the only approach assuming a statistical model where each data point is drawn from multiple components each within a different view. It has proven to successfully detect the multi-view clustering structure on various data sets.

	Multi-View	Simultaneous processing	Subspace projections	Semi-supervised	Constraint responsibility
Subspace clustering	–	✓	✓	–	–
Multi-view clustering					
↪ iterative	✓	–	–	○	fixed
↪ simultaneous	✓	✓	✓	–	–
Semi-supervised clust.	–	✓	–	✓	fixed
Our method	✓	✓	✓	✓	learned

Table 1: Overview of related paradigms

Semi-supervised clustering. As already argued, the detection of multiple clustering solutions strongly depends on the user's preferences. Semi-supervised clustering [8] provides a possibility to accommodate these preferences as additional information or domain knowledge into the clustering process. For traditional single view, full-space clustering (e.g. k-Means) a popular solution is to use instance level constraints: the objective function is extended by penalizing violated constraints [6] or one learns a distance metric that best represents the constraints [9]. For model-based clustering few extensions for equivalence constraints exist. [31] introduces a closed form EM based on the transitive closure of must-link constraints and proposes a Markov network for handling cannot-link constraints. Since it neither can incorporate both constraint types simultaneously nor cope with conflicting constraints, [25, 7] propose to integrate negative and positive pairwise constraints as priors into Gaussian mixture models, which allows for modeling soft as well as hard constraints. These approaches have shown to substantially improve the clustering result in the single view case. Since in the multi-view case we are uncertain which constraints refer to which view, these existing solutions cannot easily be transferred.

Methods such as [1] use supervision (e.g. human interaction) to enhance the clustering in a single given subspace. In contrast, we exploit supervision to enhance the clustering result across all views simultaneously. Works such as [18] combine subspace clustering with graph clustering. The underlying graph might be regarded as a certain type of supervision. These methods do not focus on finding alternative groupings in the attribute space.

Overall, none of the existing approaches is able to incorporate prior information for a multi-view clustering solution, where constraints may refer to different clustering views. Our new statistical model handles different clustering views in different attribute subspaces and learns responsibilities of views for the provided equivalence constraints.

5. EXPERIMENTAL ANALYSIS

Setup. We compare SMVC with representatives from all three paradigms: multi-view clustering, subspace clustering, and semi-supervised clustering. For multi-view clustering we choose the four approaches *Multi-View 1* and *Multi-View 2* proposed in [11], the *Alternative Clustering* method proposed in [30], and our *MVGen* [21] approach. These approaches best reflect the demands for multi-view clustering

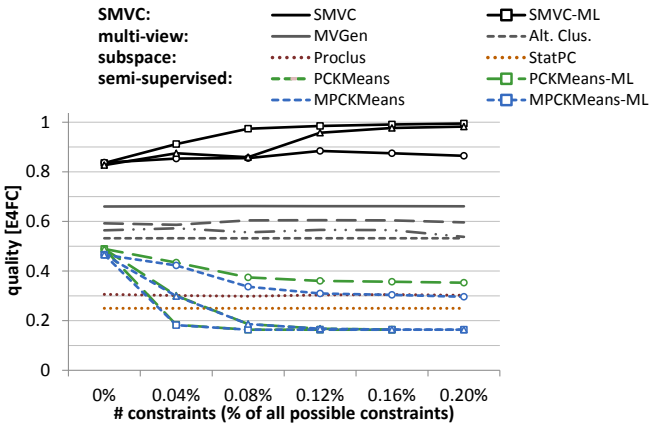


Fig. 3: Quality for a varying number of constraints

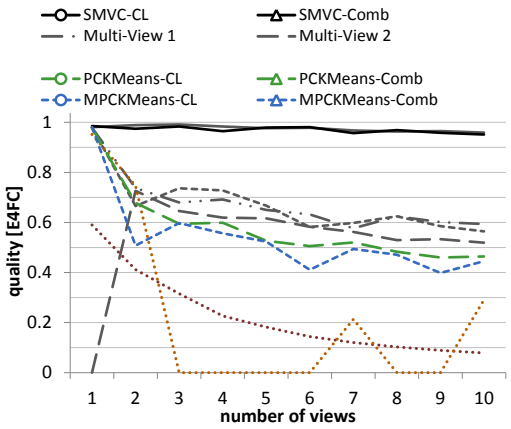


Fig. 4: Quality for a varying number of views

as discussed in Sec. 4. As subspace clustering approaches we choose the partitioning approach *Proclus* and *StatPC*, which allows for overlapping clusters. Furthermore, we compare against the two semi-supervised approaches *PCKMeans* [6] and *MPCKMeans* [9], both using instance level constraints.

For case studies on real world data we use the CMU-Faces, Iris, and Wine data (all from the UCI repository [3]), and drawn stick figures. Synthetic data containing multiple views is generated based on our generative model. The default data set contains 2 disjoint views, each with 4 clusters, 20 dimensions, and 5000 objects.

Each method is provided with the number m_{max} of views and the number k_{max} of clusters per view. If the algorithm does not allow for setting these parameters, we choose the default parameter setting.

Runtime is measured on 4GHz AMD FX-8350 CPU with 16 GB main memory. Quality is assessed based on the *E4SC* measure [20], which is a symmetric and subspace aware variant of the popular F1 measure. Since most of the competing approaches do not determine axis parallel subspaces, we refrain from evaluating the subspaces and just concentrate on the object groupings (for clarity we rename the measure to 'E4FC'). For all quality experiments, we average the results over ten executions.

5.1 Evaluation on Synthetic Data

Varying number of constraints. We start our evaluation by examining the influence of a varying number of constraints in Figure 3. Here, we tested three different variants of the semi-supervised clustering approaches: We either used only must-link constraints (SMVC-ML), only cannot-link constraints (SMVC-CL), or a combination of 50% from both (SMVC-Comb). Note that in this experiment, we randomly generated constraints based on the ground truth clusters known for synthetic data. These constraints might not help to improve the clustering and, thus, represent only very weak supervision. In practice, the user might provide better constraints, e.g. via the principles of active learning [6].

Figure 3 shows the results for an increasing number of constraints: Here, we generated a challenging dataset with a large variance to study the benefit of semi-supervision. Most approaches fail to identify a meaningful clustering structure for this difficult clustering scenario. SMVC is not only the approach showing the best clustering results without the help of prior knowledge, it is also the only approach able to improve its clustering based on additional constraints. For the two other semi-supervised approaches PCKMeans and MPCKMeans, we even observe a decreasing clustering quality with increasing amount of prior knowledge! This indicates, that they cannot deal with the potentially disagreeing constraints of the two views.

We furthermore can see the varying influence of the different constraints (100% must-link constraints, 100% cannot-link constraints, or 50%must-link + 50% cannot-link). The higher the proportion of must-link constraints, the higher is the influence. The reason is that cannot-link constraints a priori have a higher possibility to be fulfilled than must-link constraints (for m views, each with k clusters, the probability to fulfill a cannot-link constraint is $m \cdot \binom{k}{2}$, whereas for must-link constraints it is $m \cdot k$). Therefore, we will focus on must-link constraints in the following experiments.

Another interesting observation, also stated in [16], is that more constraints do not necessarily result in a better quality. They can even decrease the clustering quality. In Figure 3 we can observe this slightly for cannot-link constraints (SMVC-CL); other experiments showed similar effects for must-link constraints. We kindly refer to [16] for a discussion about these effects. Unfortunately the principles discussed in [16] for wisely choosing the set of constraints are not easily transferable to our scenario.

Varying number of views. In the next experiment, we study the potential of using SMVC as an unsupervised technique in a multi-view setting. In Figure 4, we vary the number of hidden views in the data. The dimensionality of each view is 5, i.e. with increasing number of views, the data's overall dimensionality increases as well. As depicted, SMVC and MVGen are the only approaches able to detect the clustering structure in the case of a large number of views. Their clustering quality is very high and proves to be robust against a varying number of views. The competing methods behave differently: while for single-view data the quality is relatively high, their quality heavily decreases with an increasing number of views.

Scalability. Even though the focus for SMVC lies on its clustering quality, we briefly analyze its efficiency. As already discussed in Section 3, SMVC scales linearly in the number of objects (Figure 5), linearly in the number of dimensions (Figure 6), and linearly in the number of constraints (Figure 7). Please note the logarithmic scaling of both axes in all three plots. For a varying database size

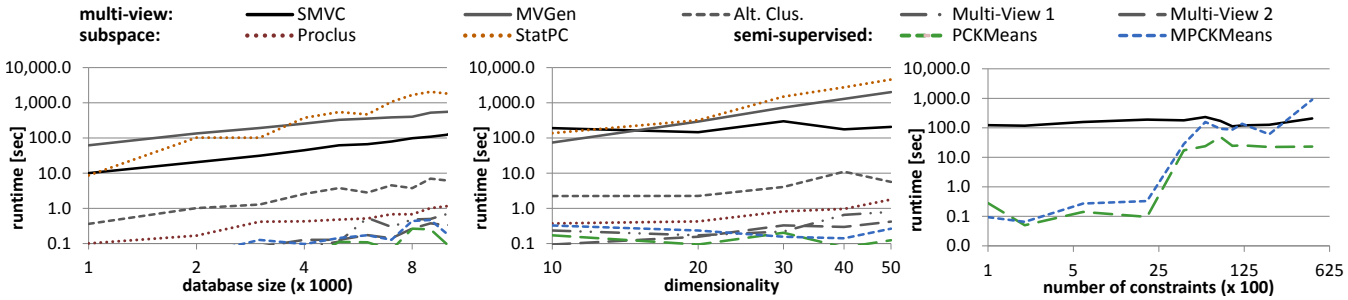


Fig. 5: Runtime vs. database size **Fig. 6: Runtime vs. dimensionality** **Fig. 7: Runtime vs. # constraints**

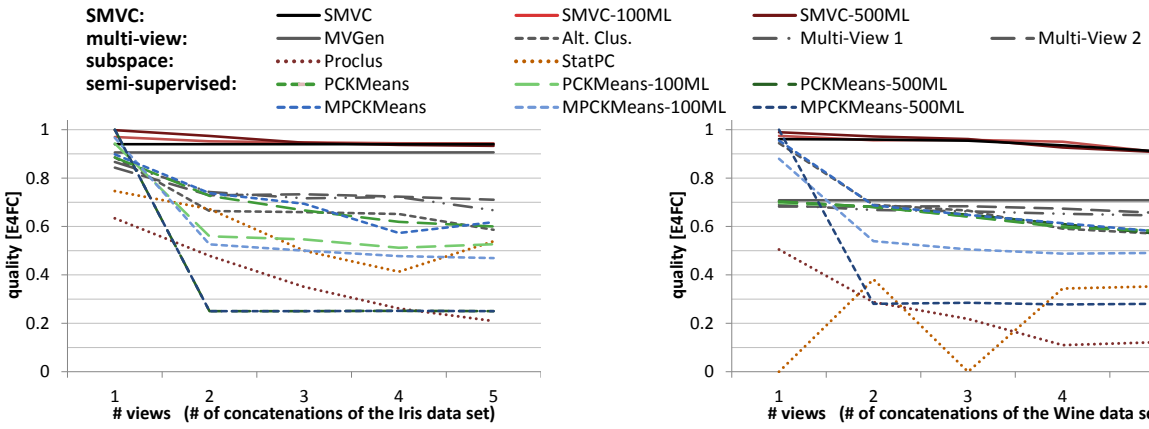


Fig. 8: Quality on iris data **Fig. 9: Quality on wine data**

(Figure 5), all algorithms show an increasing runtime. The approaches that represent adaptations of the simple and efficient KMeans algorithm (which also includes Proclus) clearly show the lowest runtimes. The runtime of SMVC is comparable to the other algorithms analyzing subspace projections (MVGen, StatPC) and even manages to outperform them thanks to the efficient variational inference techniques.

The benefit of SMVC becomes apparent for a high data dimensionality (Figure 6). Due to the exponential number of subspaces, most subspace clustering algorithms (e.g. StatPC) suffer from a tremendously increasing runtime for an increasing number of dimensions. Also MVGen cannot compete with our SMVC due to the complex model selection process. Contrarily, for SMVC, we observe a moderate increase in runtime. This enables us to apply SMVC also on high-dimensional data, as we will see in the experiments on real world data.

Figure 7 shows the runtime results of the semi-supervised methods for a varying number of constraints. Here, it is hard to verify the linear runtime of SMVC because constraints support the clustering procedure and, thus, help decreasing the number of iterations. For a small number of constraints, the two KMeans-based approaches can maintain a low runtime. For an increasing number of constraints, however, their runtime eventually even meets the one of SMVC. Of course, such a high number of constraints might not be realistic for most applications.

5.2 Evaluation on Real World Data

For evaluation on real world data we use different evaluation principles, all focusing on the multi-view aspect.

Case study A. In Figures 8 and 9, we extend the data sets Iris and Wine to data containing multiple views: for this, we randomly concatenate the attribute values of different objects up to five times to a higher dimensional space. The original data sets have dimensionalities of 4 and 13, respectively, while the extension to multi-view data leads to dimensionalities up to $5 \cdot 4 = 20$ (Iris) and $5 \cdot 13 = 65$ (Wine).

For just one view, the quality of some competing approaches is similar to the one of SMVC. However, for an increasing number of views the clustering quality for almost all competing approaches decreases. Only MVGen and SMVC are nearly not affected by an increasing number of views but detect the different object groupings even for multiple views.

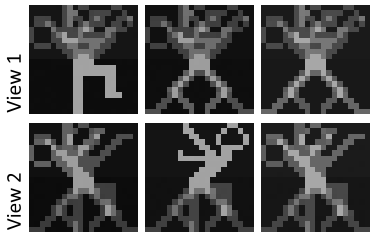
To study the effects of semi-supervision, we additionally provided for both datasets 100 and 500 constraints. For just a single view SMVC is able to improve the cluster quality. On iris, for example, the quality increases from 0.94 over 0.97 to 1.0. The full potential of our approach, however, can bee seen in the case of multiple views: While it is still able to benefit from prior knowledge, the clustering quality of the competing approaches dramatically decreases.

It is noticeable, that with increasing number of views, the constraints seem to have less positive effect on the result of SMVC. This phenomenon can, however, easily be explained by the fact that the constraints have to be distributed among the views, i.e. the proportion of prior knowledge decreases with increasing number of views.

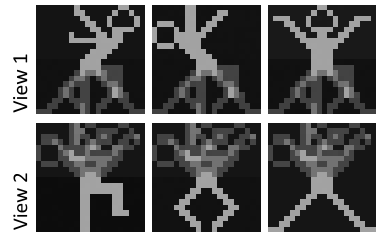
Summarizing, the results for real world data are consistent with the observations made for the synthetic data.

Case study B. For our next study, we created a data set consisting of 900 20x20 images of 'dancing stick figures'. This dataset allows an easy visual interpretation of the clustering results. We drew 9 basic stick figures (Figure 10(a)) and built 900 samples by randomly introducing noise. Since the subspace clustering and single-view clus-

(a) Samples of the stick figures data



(b) SMVC result with 0 constraints



(c) SMVC with 100 constraints

Algorithm	E4FC
SMVC 0 constraints	0.700
SMVC 100 constraints	**1**
MVGen	0.760
Alt. Clus.	0.585
Multi-View 1	0.735
Multi-View 2	0.781

(d) Multi-view algorithm results

Fig. 10: Evaluation of multi-view clustering algorithms on the stick figures dataset

(a) SMVC result with 0 constraints

(b) SMVC with 100 constraints

Algorithm	E4FC
SMVC 0 constraints	0.691
SMVC 100 constraints	**0.780**
MVGen	0.720
Alt. Clus.	0.667
Multi-View 1	0.623
Multi-View 2	0.666

(c) Multi-view algorithm results

Fig. 11: Evaluation of multi-view clustering algorithms on the faces data set

tering approaches have proven to be not applicable for the multi-view scenario, we applied only the multi-view clustering approaches in this experiment. We provide this data set on our website[1]

Although this data does not seem to be very complex, all approaches are challenged in identifying two meaningful views as shown by their clustering results (cf. Figure 10(d)). Even the initial result of our SMVC approach is not convincing as it produces the clustering depicted in Figure 10(b), which is very similar to those of the other approaches. The illustrated images correspond to the means of each detected cluster. In contrast, if we provide SMVC with 100 must-link constraints, it is able to perfectly identify the two clustering views as depicted in Figure 10(c). These two views differentiate between the stick figures' top position (view 1) and their leg position (view 2). Please note that we only choose 100 random constraints out of the 269,100 ($= 2 \cdot (3 \cdot \binom{300}{2})$) possible constraints. By exploiting this small amount of prior knowledge, our SMVC approach clearly outperforms all competing methods.

Case study C. To show that the findings of the stick figures data also apply to more complex scenarios, we next analyze the clustering result of all multi-view approaches on the CMUFace data. This data is interesting for multi-view clustering since it consists of images taken from persons showing varying characteristics such as their facial expressions (neutral, happy, sad, angry), head positions (left, right, straight, up), and eye states (open, sunglasses). As also done in [12], we randomly select 3 persons with all their images and applied PCA retaining at least 90% of the data's variance as a pre-processing.

[1]http://www.dme.rwth-aachen.de/SMVC

The result of SMVC without prior knowledge for two views each with three clusters is illustrated in Figure 11(a). The images correspond again to the clusters' means. By visual inspection, we can easily identify that the first view partitions the images based on the 3 different persons. The second view, in contrast, cannot be explained easily.

If we provide 100 constraints in order to find one view for partitioning w.r.t. the persons and another view to partition w.r.t. the head position (in total 2,592 ($= 3 \cdot \binom{32}{2} + 4 \cdot \binom{24}{2}$) possible constraints), SMVC gets the result depicted in Figure 11(b). Here we can easily identify the different head positions straight, side (left and right), and up (note that we have four head positions but only search for 3 clusters). Using the original labels provided by the dataset as ground truth, i.e. the groupings based on the different persons and the grouping based on different head positions, we obtain the clustering results of Figure 11(c). We can see, that the unsupervised multi-view approaches all yield similar clustering qualities. They were only able to identify the first view. For SMVC, we can observe a noticeable quality improvement if we integrate prior knowledge into the clustering process.

Overall, our experiments show that SMVC is able to detect the multi-view clustering structure on a variety of data sets. It successfully solves the challenge to learn the assignment of user constraints to views such that it is able to improve its clustering results based on this prior knowledge.

6. CONCLUSION

We have presented the semi-supervised clustering method SMVC that detects multiple clustering solutions in subspace projections and that exploits prior knowledge by incorporating instance level constraints. Our method is based on a sound Bayesian framework which models the data via multi-

ple mixture distributions. The model uses the instance level constraints to guide the clustering of objects, and it automatically determines which views are responsible for which constraints. For learning the clustering, we use the principle of variational inference. Our experimental study has shown the high potential of SMVC to detect multiple clustering views and its capability to use the prior knowledge for improving the clustering results.

Acknowledgments. This work has been partly funded by the SteerSCiVA DFG-664/11 project (part of SPP 1335). Stephan Günnemann has been supported by a fellowship within the postdoc-program of the German Academic Exchange Service (DAAD).

7. REFERENCES

[1] C. C. Aggarwal. A human-computer interactive method for projected clustering. *IEEE Trans. Knowl. Data Eng.*, 16(4):448–460, 2004.

[2] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. In *SIGMOD*, pages 61–72, 1999.

[3] A. Asuncion and D. Newman. UCI machine learning repository, http://archive.ics.uci.edu/ml, 2010.

[4] E. Bae and J. Bailey. Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *ICDM*, pages 53–62, 2006.

[5] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R. J. Mooney. Model-based overlapping clustering. In *KDD*, pages 532–537, 2005.

[6] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *SDM*, pages 333–344, 2004.

[7] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD*, pages 59–68, 2004.

[8] S. Basu, I. Davidson, and K. Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 2008.

[9] M. Bilenko, S. Basu, and R. J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, pages 81–88, 2004.

[10] C. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[11] Y. Cui, X. Z. Fern, and J. G. Dy. Non-redundant multi-view clustering via orthogonalization. In *ICDM*, pages 133–142, 2007.

[12] X. H. Dang and J. Bailey. Generation of alternative clusterings using the cami approach. In *SDM*, pages 118–129, 2010.

[13] X. H. Dang and J. Bailey. A hierarchical information theoretic technique for the discovery of non linear alternative clusterings. In *KDD*, pages 573–582, 2010.

[14] X. H. Dang and J. Bailey. A framework to uncover multiple alternative clusterings. *Machine Learning*, pages 1–24, 2013.

[15] X. H. Dang and J. Bailey. Generating multiple alternative clusterings via globally optimal subspaces. *DMKD*, 28(3):569–592, 2013.

[16] I. Davidson. Two approaches to understanding when constraints help clustering. In *KDD*, pages 1312–1320, 2012.

[17] Q. Fu and A. Banerjee. Multiplicative mixture models for overlapping clustering. In *ICDM*, pages 791–796, 2008.

[18] S. Günnemann, B. Boden, and T. Seidl. Finding density-based subspace clusters in graphs with feature vectors. *DAMI*, 25(2):243–269, 2012.

[19] S. Günnemann and C. Faloutsos. Mixed membership subspace clustering. In *ICDM*, pages 221–230, 2013.

[20] S. Günnemann, I. Färber, E. Müller, I. Assent, and T. Seidl. External evaluation measures for subspace clustering. In *CIKM*, pages 1363–1372, 2011.

[21] S. Günnemann, I. Färber, and T. Seidl. Multi-view clustering using mixture models in subspace projections. In *KDD*, pages 132–140, 2012.

[22] S. Günnemann, E. Müller, I. Färber, and T. Seidl. Detection of orthogonal concepts in subspaces of high dimensional data. In *CIKM*, pages 1317–1326, 2009.

[23] P. Jain, R. Meka, and I. S. Dhillon. Simultaneous unsupervised learning of disparate clusterings. *SADM*, 1(3):195–210, 2008.

[24] H.-P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *TKDD*, 3(1), 2009.

[25] Z. Lu and T. K. Leen. Semi-supervised learning with penalized probabilistic clustering. In *NIPS*, pages 849–856, 2004.

[26] G. Moise and J. Sander. Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In *KDD*, pages 533–541, 2008.

[27] E. Müller, S. Günnemann, I. Färber, and T. Seidl. Discovering multiple clustering solutions: Grouping objects in different views of the data. In *ICDM*, 2010.

[28] D. Niu, J. G. Dy, and M. I. Jordan. Multiple non-redundant spectral clustering views. In *ICML*, pages 831–838, 2010.

[29] L. K. M. Poon, N. L. Zhang, T. Chen, and Y. Wang. Variable selection in model-based clustering: To do or to facilitate. In *ICML*, pages 887–894, 2010.

[30] Z. Qi and I. Davidson. A principled and flexible framework for finding alternative clusterings. In *KDD*, pages 717–726, 2009.

[31] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall. Computing gaussian mixture models with em using equivalence constraints. In *NIPS*, 2003.

APPENDIX

For the variational distributions, the following holds:

$$\mathbb{E}_q[\![z_{m,i} = k]\!] = \psi_{m,i,k} \qquad \mathbb{E}_q[\![c_{i,j} = m]\!] = \xi_{i,j,m}$$

$$\mathbb{E}_q[\![v_d = m]\!] = \phi_{d,m} \qquad \mathbb{E}_q[\pi_{m,k}] = \frac{\tilde{\lambda}_m[k]}{\sum_{i=1}^{K} \tilde{\lambda}_m[i]}$$

$$\mathbb{E}_q[\log \pi_{m,k}] = \psi(\tilde{\lambda}_m[k]) - \psi(\sum_{i=1}^{K} \tilde{\lambda}_m[i])$$

$$\mathbb{E}_q[\mu_{m,k,d}] = \tilde{\mu}_{m,k,d} \qquad \mathbb{E}_q[\mu_{m,k,d} \cdot \tau_{m,k,d}] = \tilde{\mu}_{m,k,d} \cdot \frac{\tilde{\alpha}}{\tilde{\beta}}$$

$$\mathbb{E}_q[\tau_{m,k,d}] = \frac{\tilde{\alpha}}{\tilde{\beta}} \qquad \mathbb{E}_q[\log \tau_{m,k,d}] = \psi(\tilde{\alpha}) - \log(\tilde{\beta})$$

$$\mathbb{E}_q[\mu^2_{m,k,d} \cdot \tau_{m,k,d}] = \frac{1}{\tilde{\kappa}_{m,k,d}} + \tilde{\mu}^2_{m,k,d} \cdot \frac{\tilde{\alpha}}{\tilde{\beta}}$$