Stochastic Subspace Search for Top-K Multi-View Clustering

Geng Li
Rensselaer Polytechnic Institute
lig2@cs.rpi.edu

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

Mohammed J. Zaki
Rensselaer Polytechnic Institute
zaki@cs.rpi.edu

ABSTRACT

Finding multiple clustering solutions has recently gained much attention. Based on the observation that data is often multi-faceted, novel clustering methods have been introduced capable of detecting multiple, diverse clusterings. In this work-in-progress paper, we present a novel stochastic subspace search principle that tackles the requirements of multi-view clustering. The main idea is to consider each subspace as a state in a Markov chain and using Monte Carlo methods to sample the multi-view subspaces. By dynamically adapting the underlying probability density function we realize the generation of alternative clustering views. We present preliminary experimental results of our method and we describe future research directions.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms

Algorithms and Experimentation

1. INTRODUCTION

Traditional clustering techniques focus on finding a single grouping of the objects. Real world data, however, is often multi-faceted and allows multiple interpretations. Consequently, traditional clustering methods are often not able to uncover all structure hidden in the data. To tackle this challenge, the paradigm of multi-view/alternative clustering has been introduced. Methods from this paradigm are capable of finding alternative and diverse groupings in a single dataset.

Consider, for example, a movie database where for each movie multiple characteristics are recorded. While on the one hand a grouping based on the movies' genres might be detected, an alternative grouping might highlight the grouping according to the movies' directors, locations or cast. These different clusterings reveal different perspectives on the data and allow for an enhanced knowledge extraction. Other examples include image data, where multiple groupings might summarize different features of the images, customer data, where one grouping might represent the personal interests of the customers and another grouping the professional interests, or biological data, where multiple groupings provide different perspectives on the measurements recorded in several experiments or under different treatments.

In the last few years, various methods able to detect multiple clustering solutions have been introduced. According to [11], they can briefly be categorized into methods operating on the original (full-dimensional) dataspace [6], methods performing space transformations [5, 14], and methods analyzing (axis-parallel) subspace projections [9, 8]. In this work-in-progress paper, we describe a novel method belonging to the last category. We present a stochastic subspace search principle which is able to highlight different views in the data. The main idea is to consider each subspace as a state in a Markov chain, with transitions allowed, for example, only between similar subspaces. Starting from an initial subspace, we can then use Monte Carlo methods to perform random walks in the subspace space to sample the multi-view subspaces.

In contrast to [9], which does not explicitly model the views the clusters belong to, our method operates on the level of subspaces and, thus, directly ensures the grouping of clusters into views. Similar to [8], we also allow views to overlap, i.e. individual dimensions might belong to multiple views. While [8] proposes an extension of mixture models where the number of clusters needs to be specified for each view, we exploit the paradigm of kernel density estimation allowing us to automatically determine the number of clusters per view.

2. MULTI-VIEW SUBSPACE SEARCH

In this section, we present our multi-view subspace search principle. We assume a database $\mathcal{X}$ of $|D|$-dimensional points in the space $\mathbb{R}^{|D|}$ is given, where D represents all attributes of the dataspace. As mentioned above, we consider each subspace as a state in a Markov chain.[1] Similar to most works in the subspace clustering community, we refer to axis-parallel subspaces only. That is, a subspace $\mathbf{S}$ corresponds to a set

[1] Please note the difference to the work [12] where each state corresponds to one possible partitioning.

MultiClust '13, August 11, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2334-5/13/08 ...$15.00.

of dimensions $\mathbf{S} \subseteq D$. Thus, the Markov chain has a finite state space with $2^{|D|}$ states. We define the neighbors of each state to be its immediate subset-superset relationships.

Definition 1. *Given a subspace* $\mathbf{S}$*, the set of subspaces adjacent to* $\mathbf{S}$ *in the Markov chain search space is defined as*

$$N(\mathbf{S}) := \{\mathbf{S}' \subset \mathbf{S} \mid |\mathbf{S} \backslash \mathbf{S}'| = 1\} \cup \{\mathbf{S}' \subseteq D | \mathbf{S} \subset \mathbf{S}' \wedge |\mathbf{S}' \backslash \mathbf{S}| = 1)\}$$

Besides being efficient to compute, this definition of neighborhood additionally ensures that each state of the Markov chain has the same number of outgoing transitions.

Given this search space, our goal is to design an appropriate sampling scheme for multi-view subspace exploration. The better the clustering structure in a subspace and the more novel the information the clustering reveals, the higher should be the likelihood to sample this subspace. This task requires to solve multiple challenges: a) How to define a sound probability density function (pdf) that represents the goodness of subspaces. According to this pdf, the sampling is performed. b) How to realize that different views on the data are detected? We do not want to detect very similar subspaces that lead to redundancy. c) How to perform an efficient sampling according to both previous aspects? Analyzing all states of the Markov chain is obviously intractable. In the following sections, we discuss theses issues.

2.1 Quality of Subspaces

To evaluate the clustering structure of a subspace, we refer to the principle of non-parametric density estimation using kernel functions [16]. In particular, we focus on multiplicative kernels, i.e. kernels where the d-dimensional kernel function $\kappa(\mathbf{x})$ can be formulated as $\kappa(\mathbf{x}) = K(\mathbf{x}_1) \cdot \ldots \cdot K(\mathbf{x}_d)$ and K is a univariate kernel. Thus, given a database $\mathcal{X}$ of $|D|$-dimensional points in space $\mathbb{R}^{|D|}$, the multivariate kernel density estimator for the dataset in a specific subspace $\mathbf{S}$ is defined as

$$f^{\mathbf{S}}_{\mathcal{X}}(\mathbf{x}) \quad = \quad \frac{1}{|\mathcal{X}|} \sum_{\mathbf{y} \in \mathcal{X}} \left\{ \prod_{d \in \mathbf{S}} h^{-1}_{\mathbf{S},d} \cdot K(\frac{\mathbf{x}_d - \mathbf{y}_d}{h_{\mathbf{S},d}}) \right\} \qquad (1)$$

Instead of using a fixed bandwidth h, we use an adaptive bandwidth vector $\vec{\mathbf{h}}_{\mathbf{S}} = (h_{\mathbf{S},1}, h_{\mathbf{S},2}, \ldots, h_{\mathbf{S},m})$ in which each entry corresponds to one attribute of the m-dimensional subspace $\mathbf{S}$. This way, we account for the challenges of real world data where each attribute might show different characteristics. Note that the entry $h_{\mathbf{S},d}$ depends on two variables, the subspace dimensionality m and the currently considered dimension $d \in S$.

In our ongoing work, we use the Gaussian multiplicative kernel, i.e. the function K corresponds to the univariate Gaussian function. Using this set-up, we exploit Silverman's Rule of Thumb [16] to select the bandwidth in a m-dimensional subspace, i.e.,

$$h_{\mathbf{S},d} := \left(\frac{4}{m+2} \right)^{\frac{1}{m+4}} n^{\frac{-1}{m+4}} \sigma_d$$

where σ_d denotes the standard deviation for the d-th attribute and $m = |\mathbf{S}|$. The rationale behind choosing Silverman's Rule of Thumb is when using the Gaussian multiplicative kernel, the asymptotic mean integrated squared error (AMISE) is minimized [15]. It is fair to mention that this adaption provides only a rough estimate for the bandwidth. In general, kernel density estimation for multi-dimensional data is a challenging task and different methods for bandwidth selection have been proposed including methods using a variable bandwidth over the domain to be estimated [17]. The above rule, though, does a first step in this direction while simultaneously allowing an efficient computation.

Additionally, to ensure the comparison of the density values between subspaces of different cardinality, we normalize the density of a point $\mathbf{x}$ by $f^{\mathbf{S}}_{\mathcal{X}}(\mathbf{x}) \leftarrow f^{\mathbf{S}}_{\mathcal{X}}(\mathbf{x}) \cdot vol(\mathbf{S})$ where $vol(\mathbf{S})$ is defined as the volume of the hypersphere which encloses the datapoints in subspace $\mathbf{S}$. This principle follows the idea of an dimensionality unbiased computation as, e.g., proposed in [2]. The multiplication by $vol(\mathbf{S})$ avoids the bias to lower-dimensional subspaces. While this solution has shown good results in our experiments, we are currently studying further approaches how to ensure a fair comparison between the density values of different subspaces. Similar to the problem of bandwidth selection discussed above, it might be reasonable to replace the global normalization of the overall subspace by a variable normalization which considers local properties of specific regions.

Assessing the clustering structure.

The function $f^{\mathbf{S}}_{\mathcal{X}}$ provides a density estimate for the entire dataset, including potentially sparse regions of the data. For our method, however, we are only interested in assessing the clustering structure of the corresponding subspace. Thus, instead of measuring the density of the entire dataset, we measure the density only w.r.t. the hidden clusters. For detecting the clusters, we use the well established Mean Shift clustering method [4] in combination with the kernel function introduced above. By using Mean Shift we are able to detect arbitrarily shaped clusters and we avoid setting the number of clusters as a predefined parameter, thus, allowing us to detect views where the number of clusters varies.

Overall, by replacing in Equation 1 the set $\mathcal{X}$ with the individual clusters C_i, we formalize:

Definition 2. *Let* $\mathcal{C}_S = \{C_1, \ldots, C_m\}$ *be the set of clusters in subspace* $\mathbf{S}$*. The probability density function* $Q(\mathbf{S})$ *which reflects the clustering structure of each subspace is defined as*

$$Q(\mathbf{S}) \propto \frac{1}{|\mathcal{X}|} \sum_{C_i \in \mathcal{C}_S} |C_i| \cdot \sum_{\mathbf{x} \in C_i} f^{\mathbf{S}}_{C_i}(\mathbf{x})$$

Intuitively, the function $f^{\mathbf{S}}_{C_i}$ provides a density estimate for the cluster C_i only (in terms of mixture models, this function would correspond to a single component of the mixture distribution). We then compute the weighted average over all clusters according to their cluster sizes (similar to mixture models where the weighted sum of the components is computed). Since the function $f^{\mathbf{S}}_{C_i}$ is evaluated at the corresponding datapoints, the measure $Q(\mathbf{S})$ well assesses the quality of the subspace.

Please note that the function $f^{\mathbf{S}}_{\mathcal{X}}(\mathbf{x})$ as well as the function $Q(\mathbf{S})$ represent probability density functions. However, they operate on completely different domains. The first one is used to estimate the distribution of the points in the corresponding subspace. That is, the function's domain are points in a $|\mathbf{S}|$-dimensional space. The function $Q(\mathbf{S})$, in contrast, assesses the clustering structure of the subspaces. Its domain corresponds to all possible subspace projections.

2.2 Top-K Multi-View Subspaces

Since very similar subspaces often show similar clustering structure and, thus, similar quality, simply sampling from the above pdf does not meet the needs for detecting alternative views. To solve this issue, we exploit the following idea: We do not use a static pdf, but we adapt the pdf during the clustering process. After detecting a good subspace **S**, we lower the likelihood of sampling similar subspaces. The pdf will be *locally* distorted.

Formally, this local distortion is defined as:

DEFINITION 3. *Given a set of already detected subspaces* $\mathcal{M} = \{\mathbf{S}_1, \ldots, \mathbf{S}_q\}$*, the adapted pdf is defined as:*

$$Q(\mathbf{S}|\mathcal{M}) \propto (1 - sim(\mathbf{S}, \mathcal{M})) \cdot Q(\mathbf{S})$$

where $sim(\mathbf{S}, \mathcal{M}) := \max_{\mathbf{S_i} \in \mathcal{M}}(\frac{|\mathbf{S}_1 \cap \mathbf{S}_2|}{|\mathbf{S}_1 \cup \mathbf{S}_2|})$

The more similar a subspace **S** is to one of the already detected subspaces from $\mathcal{M}$, the stronger the pdf is adapted, i.e. the lower the subspace's probability. In the case of $\mathbf{S} \in \mathcal{M}$, we get $Q(\mathbf{S}|\mathcal{M}) = 0$; thus, preventing to sample the same subspace twice. Subspaces that are not located in the local neighborhood of any previously detected subspace are not affected at all.

Based on this adaptive pdf, our overall goal can be formulated as:

DEFINITION 4. *Top-k multi-view subspaces*
Finding the top-k multi-view subspaces corresponds to determining subspaces $\mathbf{S}_1, \ldots, \mathbf{S}_k$ *such that*

$$\forall i: \; \mathbf{S}_i = \operatorname*{argmax}_{\mathbf{S} \subseteq D} Q(\mathbf{S}|\mathcal{M}_{i-1})$$

where $\mathcal{M}_i = \{\mathbf{S}_1, \ldots, \mathbf{S}_i\}$ *and* $\mathcal{M}_0 = \emptyset$*.*

2.3 Efficient Sampling Schemes

Analyzing all subspaces to find the top-k result is obviously intractable. Instead we propose approximate methods based on Monte Carlo sampling to determine the solution.

Simulated Annealing (SA).

Our first principle to sample subspaces follows a simulated annealing approach [18]. SA is an optimization method for finding a good approximation of the global optimum of a target function. SA was inspired from thermodynamic simulation, which involves heating and controlled cooling of a material in order to reduce the defect rate. Here, we sample a subspace **S** from the space of all possible subspaces according to the rule

$$p(\mathbf{S}) \propto \exp\left\{\frac{Q(\mathbf{S}|\mathcal{M})}{T}\right\}$$

where $T > 0$ is the temperature. By successively lowering the temperature T, the generated samples will concentrate around the maximum of the function $Q(\mathbf{S}|\mathcal{M})$.

We use the above equation within a Metropolis-Hastings algorithm. Thus, given the current state of the Markov chain, i.e. the subspace **S**, the acceptance probability for a random $\mathbf{S}' \in N(\mathbf{S})$ is:

$$\begin{aligned} \mathcal{A}_{SA}(\mathbf{S}, \mathbf{S}') &= \min\left\{1, \frac{\exp\left\{\frac{Q(\mathbf{S}'|\mathcal{M})}{T}\right\} \cdot |N(\mathbf{S})|}{\exp\left\{\frac{Q(\mathbf{S}|\mathcal{M})}{T}\right\} \cdot |N(\mathbf{S}')|}\right\} \\ &= \min\left\{1, \exp\left\{\frac{\Delta Q(\mathbf{S}', \mathbf{S}|\mathcal{M})}{T}\right\}\right\} \\ &= \begin{cases} 1 & \text{, if } Q(\mathbf{S}'|\mathcal{M}) > Q(\mathbf{S}|\mathcal{M}) \\ \exp\left\{\frac{\Delta Q(\mathbf{S}', \mathbf{S}|\mathcal{M})}{T}\right\} & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

where $\Delta Q(\mathbf{S}', \mathbf{S}|\mathcal{M}) = Q(\mathbf{S}'|\mathcal{M}) - Q(\mathbf{S}|\mathcal{M})$. The higher the temperature T the more likely we accept a subspace with lower quality, enabling the algorithm to move off a local maxima. Subspaces with better quality are always accepted.

Algorithm 1: Simulated Annealing (SA) exploration

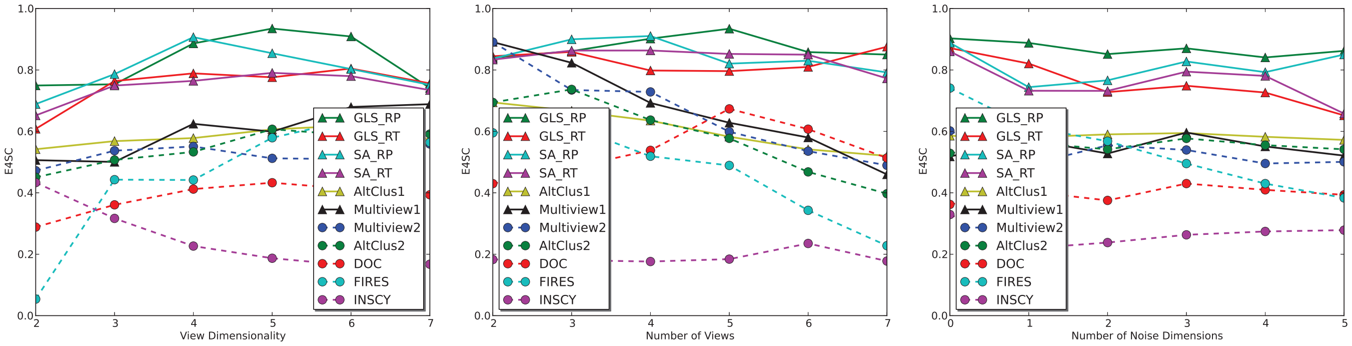
```
   Input: $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, $k$, $T_0$, $\alpha$
   Output: Multiview Subspaces $\mathcal{M}$
1  $T \leftarrow T_0$;
2  $\mathbf{S} \leftarrow$ Init-Subspace($\emptyset$);
3  while $|\mathcal{M}| \neq k$ do
4      Choose $\mathbf{S}'$ uniformly from $N(\mathbf{S})$;
5      Accept $\mathbf{S}'$ with probability as given in Eq. 2;
6      if $\mathbf{S}'$ is accepted then $\mathbf{S} \leftarrow \mathbf{S}'$;
7      ;
8      else // new subspace is rejected
9          if $T < 0.001$ then
10             Remove $\mathbf{S}'$ from $N(\mathbf{S})$;
11             if $N(S) == \emptyset$ then
12                 Insert $\mathbf{S}$ in $\mathcal{M}$;
13                 $\mathbf{S} \leftarrow$ Init-Subspace($\mathcal{M}$);
14                 $T \leftarrow T_0$;
15     UpdateT($T$, $\alpha$);
```

Algorithm 1 outlines the main idea of the SA-based exploration for multi-view subspace detection. We start with an initial subspace (line 2; cf. details at the end of this section). We then (line 4) uniformly choose a neighbor $\mathbf{S}'$ from the current subspace's neighborhood, and we accept it with probability according to Equation 2 (line 5). If the new subspace is accepted we set it as the current state of the Markov chain. If the chosen neighbor is rejected, we stay at the current subspace. To avoid repeated sampling of unpromising subspaces, we additionally remove rejected subspaces from the current subspace's neighborhood (line 9). Please note that this step is only performed when the temperatures has reached a small value. For large T values, unpromising subspaces are retained, thus, allowing us to explore a larger part of the search space. If all neighbors of a subspace **S** have been removed (line 10), we have reached a local maxima. Thus, we add the subspace to the current result set $\mathcal{M}$ and we start a new random walk in the Markov chain.

Finally, in line 14 we realize the lowering of the temperature T as required for the simulated annealing method. We adopt the commonly used cooling scheme based on the geometric rule for temperature variation [18], i.e. $T_{j+1} = \alpha \cdot T_j$ where α is a positive constant smaller than 1. Typical values for α are in the range of 0.80 to 0.99 [1].

Greedy Local Search (GLS).

While the SA method allows to accept neighboring subspaces showing lower quality, we additionally analyze a sec-



(a) Varying View Dimensionality

(b) Varying Number of Views

(c) Varying Noise Dimensionality

Figure 1: Experiments on Synthetic Datasets

ond principle based on a greedy local search: Given the current subspace $\mathbf{S}$, we do not perform a random transition (lines 4-5 of Alg. 1) but the next state of the Markov chain is chosen as

$$\mathbf{S}' = \operatorname*{argmax}_{\widehat{\mathbf{S}}\in N(\mathbf{S})} \left\{ Q(\widehat{\mathbf{S}}|\mathcal{M}) \mid Q(\widehat{\mathbf{S}}|\mathcal{M}) > Q(\mathbf{S}|\mathcal{M}) \right\} \qquad (3)$$

If there exists no $\widehat{\mathbf{S}} \in N(\mathbf{S})$ such that $Q(\widehat{\mathbf{S}}|\mathcal{M}) > Q(\mathbf{S}|\mathcal{M})$, we have reached a local maximum and we perform the same steps as in lines 11 and 12 of Algorithm 1.

Please note the the above principle finds a sequence of local maxima, while Definition 4 is formulated as a sequence of global maxima. Why do we expect this principle to work well? The intuition is as follows: Due to the local distortions performed within the function $Q(\mathbf{S}|\mathcal{M})$, subspaces which have been rated as local maxima in early iterations might evolve to global maxima in later iterations. This effect results due to the fact that the probability of subspaces close to the global maximum is lowered as defined by Definition 3. Thus, by directly focusing on the local maxima of the pdf, the greedy local search might well approximate the overall task.

Generating Initial Subspaces.

The choice of subspaces where the Monte Carlo sampling starts (line 2 and 12 of Algorithm 1) is crucial for a good subspace exploration. In the best case, we select subspaces which have previously not been analyzed by our method; thus, generating alternative views on the data. We exploit two strategies to select these subspaces.

Our first strategy is based on (weighted) random projection, denoted as RP. We perform an independent sampling among all attributes to determine which ones are included in the initial subspace. That is, we assign a weight $w(d)$ to each attribute $d \in D$, where $w(d)$ is the probability that d belongs to the initial subspace and $1 - w(d)$ the probability that d does not belong to the initial subspace. At the start of the algorithm, we assign to each attribute the probability $w(d) = \frac{1}{k}$, where k is the input parameter specified by the user denoting the desired top-k multi-view subspaces. Thus, the expected dimensionality of the starting subspace is equal to $\frac{|D|}{k}$. Each time a new subspace is added to the result set $\mathcal{M}$, i.e. before restarting the search for a new subspace, we update the weights $w(d)$ based on the already detected subspaces, thus, steering the search to novel solutions. More precisely, we use the weights $w(d) = \left(\frac{1}{k}\right)^{x_d+1}$, where x_d is the number of times the dimension d has already appeared in previously detected subspaces, i.e. $x_d = |\{\mathbf{S} \in \mathcal{M} | d \in \mathbf{S}\}|$. The more often a dimension d has been selected, the lower the likelihood of being selected as a dimension of the initial subspace.

While our first strategy performs an independent sampling of all dimensions $d \in D$ (thus, leading to initial subspaces of potentially more than one dimension), our second strategy is to always start from a one-dimensional subspace. The probability of the attribute $d \in D$ to be chosen as the initial subspace is selected as $\hat{w}(d) = \frac{w(d)}{\sum_{d' \in D} w(d')}$. Also for this method, we update the weights $w(d)$ as described above. We call this strategy RT for short.

3. PRELIMINARY EXPERIMENTS

To analyze our method, we have performed preliminary experiments on synthetic data containing multiple views. The default dataset contains 4 views each located in a subspace with 5 dimensions. Additionally, we added two dimensions showing no clustering structure, thus, leading to a default dataset with 22 dimensions. The number of clusters in each view is 2, 4, 6 and 8, respectively. Each cluster follows a multivariate normal distribution. The dataset has 3000 points by default. For evaluating clustering quality, we use the E4SC measure [7]. The range of E4SC is in $[0, 1]$ and the higher the E4SC measure, the better the subspace clustering result.

We denote our Simulated Annealing method as SA_RP and SA_RT where the suffix indicates the principle which is used to generate the initial subspaces. Accordingly, our Greedy Local Search is denoted as GLS_RP and GLS_RT. Since our methods are randomized, we repeat them 10 times, and report the average. For SA_RP, we set the initial temperature to $T_0 = 50$ and the cooling rate to $\alpha = 0.8$.

We compared our methods with the subspace clustering techniques DOC [13], FIRES [10] and INSCY [3]. DOC is a Monte Carlo method, and FIRES and INSCY are all density-based methods. Since DOC is a Monte Carlo method, we ran DOC as many times as the number of ground truth views. Then we combined the detected subspace clusters by DOC from all runs as the final result. For competing multi-view methods we selected Multivew1 and Multivew2 proposed in [5] and two variants AltClus1 and AltClus2 of the Alternative Clustering methods proposed in [14]. Since these methods do not generate subspace information, we ignored the subspaces during the evaluation. In other words, we evaluated only the point groupings and assume that all clusters are located in the "correct" subspaces. This way the competing multi-view methods have a huge advantage. However, as we will see in the following, the results show that the competing multi-view methods still have low quality.

Figure 1 shows the E4SC values for each algorithm when varying different characteristics of the data. In Figure 1(a), we varied the number of dimensions per view. As indicated, each of our methods outperforms the competing approaches in detecting the multi-view subspaces. The methods based on the independent random projection principle (*_RP) perform in most cases slightly better than the methods using initial subspaces of cardinality one (*_RT). For this dataset, both sampling schemes perform very similar.

Since our methods are randomized, we additionally computed the standard deviations of the obtained results. In most cases, the standard deviation was around 0.1 with most extreme values of 0.03 and 0.16 obtained by the GLS_RP method for the setting of 5-dimensional views and 2-dimensional views, respectively. Thus, even when taking the standard deviation into account, in most cases our methods obtain the best clustering results.

In Figure 1(b) we varied the number of total views in the data. While for a small number of views, some competing approaches obtain similar quality values, our methods clearly outperform them when the data contains a large number of views. Our methods show constantly high quality, while the competing approaches show strongly decreasing quality. In this experiment, the difference between the four variants of our method is not as clear as before. The methods based on random projection (RP) show a slightly better performance.

Finally, in Figure 1(c), we varied the number of irrelevant dimensions. As describe in the experimental setup, for each dataset we added a certain number of dimensions showing no clustering structure. As shown, our methods can handle data showing this characteristic. In this experiment, the GLS_RP method obtains the best results for any number of noise dimensions.

Overall, these preliminary results show that our methods can detect multiple views in various settings.

4. CONCLUSION AND FUTURE WORK

We have introduced our ongoing work for multi-view subspace exploration. Our method exploits the idea of Markov Chain Monte Carlo sampling where subspaces are sampled according to their clustering structure as well as dissimilarity to previously detected subspaces. We have analyzed two different sampling schemes based on simulated annealing and using a greedy local search as well as two different strategies to generate the initial subspaces. Overall, the proposed variants are able to detected subspaces that highlight different views on the data.

As this is preliminary work, there are still many open challenges. First, our current method is limited to find multi-view clusterings where the clusters of a single view are located in exactly the same subspace. An interesting extension would be to allow individual clusters to slightly deviate from their view, thus, realizing a more flexible detection. Second, the Mean Shift clustering algorithm used in our method is computationally intensive making an application on large datasets challenging. We plan to analyze whether alternative methods can serve as a good substitute for Mean Shift to estimate the quality of the subspaces. Third, while the kernel density estimation principle used in this paper has empirically shown to perform well, we want to analyze it more theoretically. Particularly, it is still an open challenge to derive a sound principle ensuring a fair comparison of density values among different subspaces. Finally, while currently the number of views is given as an input parameter, we want to develop a method capable of automatically determining the correct number of views. The degree of distortion of the probability density function might be a good indicator for this task. Tackling all these issues is our future research direction.

5. ACKNOWLEDGMENT

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

6. REFERENCES

[1] E. Aarts and J. K. Lenstra. *Local search in combinatorial optimization*. John Wiley & Sons, Inc., 1997.

[2] I. Assent, R. Krieger, E. Müller, and T. Seidl. Dusc: Dimensionality unbiased subspace clustering. In *ICDM*, pages 409–414, 2007.

[3] I. Assent, R. Krieger, E. Muller, and T. Seidl. Inscy: Indexing subspace clusters with in-process-removal of redundancy. In *ICDM*, pages 719–724, 2008.

[4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 24(5):603–619, 2002.

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

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

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

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

[9] 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.

[10] H. Kriegel, P. Kroger, M. Renz, and S. Wurst. A generic framework for efficient subspace clustering of high-dimensional data. In *ICDM*, pages 8–pp, 2005.

[11] E. Müller, S. Günnemann, I. Färber, and T. Seidl. Discovering multiple clustering solutions: Grouping objects in different views of the data. In *ICDE*, pages 1207–1210, 2012.

[12] J. M. Phillips, P. Raman, and S. Venkatasubramanian. Generating a diverse set of high-quality clusterings. In *MultiClust Workshop at ECML PKDD*, 2011.

[13] C. Procopiuc, M. Jones, P. Agarwal, and T. Murali. A monte carlo algorithm for fast projective clustering. In *SIGMOD*, pages 418–427, 2002.

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

[15] D. Scott. *Multivariate density estimation: theory, practice, and visualization*, volume 275. John Wiley & Sons, 1992.

[16] B. Silverman. *Density estimation for statistics and data analysis*, volume 26. Chapman & Hall/CRC, 1986.

[17] G. R. Terrell and D. W. Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

[18] P. van Laarhoven and E. Aarts. *Simulated annealing: theory and applications*, volume 37. Springer, 1987.