Outlier Ranking via Subspace Analysis in Multiple Views of the Data

Emmanuel Müller[●○] Ira Assent[◇] Patricia Iglesias[●] Yvonne Mülle[●] Klemens Böhm[●]

[●] *Karlsruhe Institute of Technology, Germany*
{*emmanuel.mueller, patricia.iglesias, klemens.boehm*}*@kit.edu and yvonne.muelle@student.kit.edu*

[○] *University of Antwerp, Belgium*
emmanuel.mueller@ua.ac.be

[◇] *Aarhus University, Denmark*
ira@cs.au.dk

***Abstract*—Outlier mining is an important task for finding anomalous objects. In practice, however, there is not always a clear distinction between outliers and regular objects as objects have different roles w.r.t. different attribute sets. An object may deviate in one subspace, i.e. a subset of attributes. And the same object might appear perfectly regular in other subspaces. One can think of subspaces as multiple views on one database. Traditional methods consider only one view (the full attribute space). Thus, they miss complex outliers that are hidden in multiple subspaces.**

In this work, we propose *OutRank*, a novel outlier ranking concept. *OutRank* exploits subspace analysis to determine the degree of outlierness. It considers different subsets of the attributes as individual outlier properties. It compares clustered regions in arbitrary subspaces and derives an outlierness score for each object. Its principled integration of multiple views into an outlierness measure uncovers outliers that are not detectable in the full attribute space. Our experimental evaluation demonstrates that *OutRank* successfully determines a high quality outlier ranking, and outperforms state-of-the-art outlierness measures.

***Keywords*-outlier ranking, multiple subspaces, clusterings**

I. Introduction

Outlier ranking is an important data mining task for the identification of anomalous, suspicious, and rare objects in large data volumes. Many applications in science and business routinely collect huge amounts of data. In practice, many of these processes face data quality issues. Sensors might fail to deliver correct values, experimental conditions may vary unpredictably, or human behavior is simply unexpected. However, due to the amount of information measured in various attributes, there is not always a clear distinction between outliers and regular objects. An object (e.g. a person o_2 in Fig. 1) might show different roles in the data, clustered w.r.t. some attributes (regular sports and sleeping behavior) while being an outlier in some other attributes (highlighting an unexpected social attitude). While existing outlier ranking approaches have been successful in detecting outliers that are relatively simple, namely those that show when considering all attributes simultaneously, they have not addressed the issue of identifying outliers that are anomalous only in some subsets of the given attributes (so-called *subspaces*).

A recent research direction has focused on data analysis in such subspaces. A broad set of clustering algorithms has been proposed for subspace cluster detection in arbitrary views of the data space [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. They form a well established research area with scalable processing schemes and various clustering models taking different application demands into account. In general, subspace clustering selects a set of relevant attributes for each cluster. It is able to detect multiple views on the same database, and groups each object accordingly to multiple subspace clusters. This type of multiple cluster assignment has shown high quality results for example in gene expression data [5], where each gene has multiple functional roles that can be detected by multiple subspace clusters. However, all of these techniques focus on object groupings and are not able to assess the deviation of individual outliers.

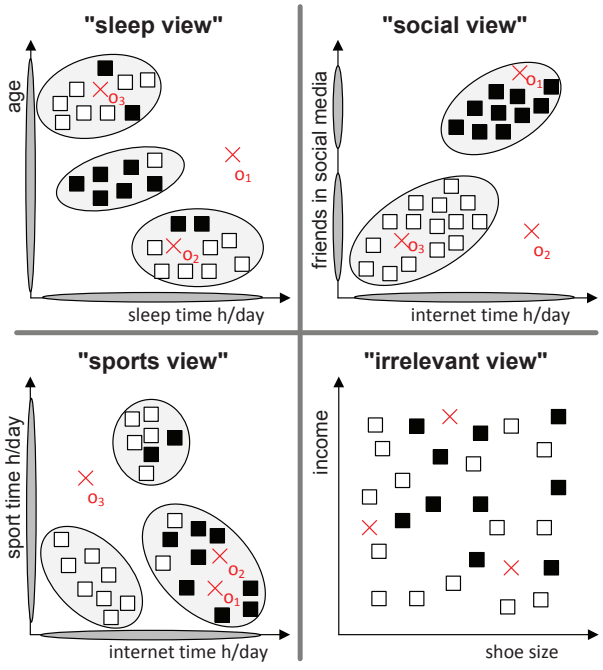


Figure 1. Outliers w.r.t. multiple subspace views

There is only relatively little research for outlier detection in subspaces [11], [12], [13]. All three of these approaches re-invent subspace analysis in the context of outlier mining and fall prey to well-known scalability and subspace selection challenges. We see huge potential in utilizing established subspace analysis models from the domain of subspace clustering for subspace outlier mining. Both efficiency and quality improvements in clustering could be exploited for subspace outliers in a general framework. However, subspace clustering poses two unique challenges for outlier detection (illustrated in Figure 1): first, each object, even if it deviates substantially in some subspaces, is very likely to be part of at least some clusters in other projections. Thus, outliers are not simply non-clustered objects. Second, assessing the degree of deviation is not straightforward. Subspace clusters represent groups of data in very many different (or similar) views, which makes the assessment of deviation a non-trivial task. An outlierness score for meaningful ranking requires a principled integration of these multiple views.

In this work, we propose a novel outlier scoring concept based on subspace analysis, following our workshop vision paper [14]. With *OutRank* we focus on the development of novel scoring functions that incorporate the extracted information represented by subspace clusters. This general concept of incorporating subspace clusters as information into outlier ranking has not yet been investigated in literature. With our scoring, we have to define novel indications for outliers only based on subspace clusters as a pre-processing result. Please note that these indications are not as simple as in traditional clustering: "object is not clustered $\Rightarrow$ it is an outlier". As each object may be clustered in multiple views, we utilize this property to extract evidence for the regularity of objects. Our general hypothesis is that regular objects show clustered behavior in multiple subspaces even if the subspaces are very dissimilar to each other. In contrast, outliers are clustered in some subspaces but deviate from these clusters if one considers other subspaces. As most prominent indication for outliers, we observe situations where objects deviate from the subspace cluster they are usually grouped in. For example, object o_2 in Fig. 1 is clustered in two views, but not in the "social view". Although there is a very similar clustering structure of the black objects in the "sports view", we observe this object deviating from its common grouping. Our *OutRank* method takes this similarity of subspaces into account, and computes the outlierness degree based on the information available from subspace analysis. In particular, we utilize the multiple cluster assignment provided by subspace clustering algorithms. We distinguish between redundant (similar) subspace views and multiple clusterings in orthogonal (dissimilar) subspaces. We demonstrate empirically that *OutRank* outperforms existing outlierness measures, we show its scalability, and evaluate its flexibility w.r.t. different subspace clustering models.

II. Related Work

In this section, we review existing techniques for outlier mining in pairs of opposed approaches, and highlight our research focus w.r.t. these directions.

Binary Detection vs. Outlier Ranking:
Originally, outlier mining meant binary detection of outliers (as opposed to inliers) [15], [16]. Also, some clustering techniques are capable of finding a set of non-clustered outliers [17]. By contrast, recent approaches determine a degree of outlierness, such as the local deviation of objects [18], which has been recently extended to high dimensional data [19], [20]. In this work, we follow the ranking paradigm which provides a more detailed view on the deviation of outliers. Please note that a binary outlier set can be easily derived through thresholding out of a ranking result.

Full Space vs. Subspace Outliers:
Traditional methods determine outliers or compute the outlier score in the full attribute space [18], [21], [19]. As such, they consider outliers to appear w.r.t. all of the given attributes, and fail to uncover complex deviation hidden in some attributes only. In the literature, a binary subspace outlier method [11] and ranking in subspaces [13], [12] have been proposed. In a sense, these approaches re-invent selection strategies for subspaces in the context of outlier mining. They ignore recent results in subspace clustering, both in terms of efficiency and subspace selection. With our work we bridge this gap and exploit the advances in subspace clustering research for the benefit of complex outliers hidden in multiple subspaces.

A Single Projection vs. Multiple Subspaces:
Well-established approaches, such as principal components analysis [22], can be used to reduce the data space to a single projection. However, such a projection of the entire data space is not aware of individual projections for subsets of the objects. A single projection misses different views of the data. A first technique uses multiple random projections to cope with this issue [23]. However, this means that relevant structures from which outliers deviate can be missed in (random) non-selected subspaces. In contrast to single and random projections, subspace clustering detects clusters in any possible combination of attributes. It covers a variety of approaches in subspace clustering [1], [4], [5], projected clustering [2], [3], and non-redundant subspace clustering [6], [7], [8]. Details can be found in a recent survey [9] and an evaluation study [10]. *OutRank* builds on these successful subspace analysis results, but does not assume a particular method. It uses subspace clustering as a meaningful preprocessing step for subspace and cluster detection. We fully explore this idea and provide novel scoring functions that assess the evidence in the entire subspace clustering result.

III. Outlier Ranking via Subspace Analysis

The goal in *OutRank* is to derive a ranking of all objects in the database w.r.t. to their deviation from the remainder of

the data. In order to assess also complex deviations, subspace clusters are analyzed, and the results are integrated into a score for each object. Subspace clustering provides groups of regular objects, and potential outliers in the respective subspace. To compute the score, we have to formalize the degree of regularity (or deviation) of an object in the subspace and how to integrate these partial scores to derive the overall score. An important consideration for the integration is to avoid bias associated with similar views that do not carry new information regarding regularity (or deviation) of an object.

A. Basic Notions

Formally, for a d-dimensional database DB with each object o described by a vector $(o_1, \ldots, o_d) \in \mathbb{R}^d$ in the full space $\mathcal{D} = \{D_1 \ldots D_d\}$, *OutRank* computes a $score(o) \in [0 \ldots 1]$ for each object $o \in DB$. $score(o)$ represents the degree of *regularity*, thus $1 - score(o)$ is the *outlierness degree*. This means that perfect inliers score close to a value of 1, and highly deviating outliers score close to a value of 0. The outlier ranking is simply a sorted list of DB in *ascending* order of $score(o)$.

In order to find complex deviations, i.e., deviations that are not visible in the full space, we analyze subspaces. Each subspace is a *view* of the data in which we determine regular data, and deviations. Formally, a subspace is defined as a subset of the given attributes:

Definition 1: Subspace S
Given the full data space $\mathcal{D} = \{D_1 \ldots D_d\}$, a subspace is defined as:

$$S \subseteq \mathcal{D}$$

Obviously, there are far too many subspaces to explore. With increasing number of attributes d, we observe an exponential increase of $2^D - 1$ possible subspaces. Selecting the relevant subspaces for each cluster or outlier is a main research goal. Each subspace S represents a different view on DB, hence, distances are restricted to $D_i \in S$. For example, the restricted Euclidean distance in S is defined as:

$$dist_S(o, p) = \sqrt{\sum_{D_i \in S} (o_i - p_i)^2}$$

Based on this definition of subspaces, a variety of subspace clustering algorithms have been proposed. Each technique provides a different cluster definition (e.g. grid-based [1], [4], density-based [5], [6], ...), which fulfills a certain application demand. We abstract from these individual definitions and use their general clustering result as input for our outlier ranking.

At a very general level, we provide an abstract definition of a scoring function, given a subspace clustering, as follows:

Definition 2: Basic Outlier Scoring
Let $SCR = \{(C_1, S_1), \ldots, (C_k, S_k)\}$ a subspace clustering result, i.e., a set of clusters C_i in their associated subspaces S_i. A scoring function on SCR is then defined as:

$$score(o) = \sum_{\{(C,S) \in SCR \mid o \in C\}} evidence(o, (C, S), SCR)$$

where $evidence$ computes a value of regularity for o being clustered in subspace cluster (C, S) given the entire subspace clustering result SCR.

This definition provides the abstraction of our framework, and does not put any restrictions to SCR or the underlying subspace clustering approach. The abstract notion "evidence of regularity" will be instantiated by concrete *OutRank* scoring functions in the following (cf. Section III-C). Nonetheless, *OutRank* as a framework does not require a particular instantiation, and can therefore be adapted to new developments in scoring or subspace analysis. We consider this decoupling of *scoring* and *subspace analysis* as a major contribution to the development of future outlier ranking techniques.

B. Challenges with Subspace Analysis

In the simplest case, an object o might not appear in any subspace cluster, and yields $score(o) = 0$. In practice, however, an object is typically clustered (differently) in multiple subspaces, and it might show different degrees of regularity/deviation in different subspaces. We therefore need to carefully design a principled way to assess the information provided by different subspaces and how they relate to one another.

As we can immediately see from the restricted distance function $dist_S$, objects typically show different behaviors in multiple views S. With *OutRank*, we rely on the general observation that outliers are objects which do not agree with other data in at least some of the attributes:

(1) Outliers may be regular in some subspaces
(2) They deviate in at least some subspaces
$\Rightarrow$ Assessment of different subspaces indicates their outlierness

Since subspaces may contribute conflicting or redundant information, *OutRank* assesses not only the regularity in each view, but also takes the remaining views into account. *OutRank* is the first outlier mining approach that tackles these challenges for an outlier analysis based on such multiple views.

Challenge 1: Multiple Views
Multiple views, as uncovered by subspace clustering, render binary scoring meaningless, i.e., a scoring function as follows

$$score(o) = \begin{cases} 1 & , \text{ if } \exists (C, S) \in SCR \ \wedge \ o \in C \\ 0 & , \text{ else.} \end{cases}$$

is expected to find few, if not none, outliers in practice.
Challenge 1 is a direct consequence of the observation discussed above: as typically all objects o are part of at least

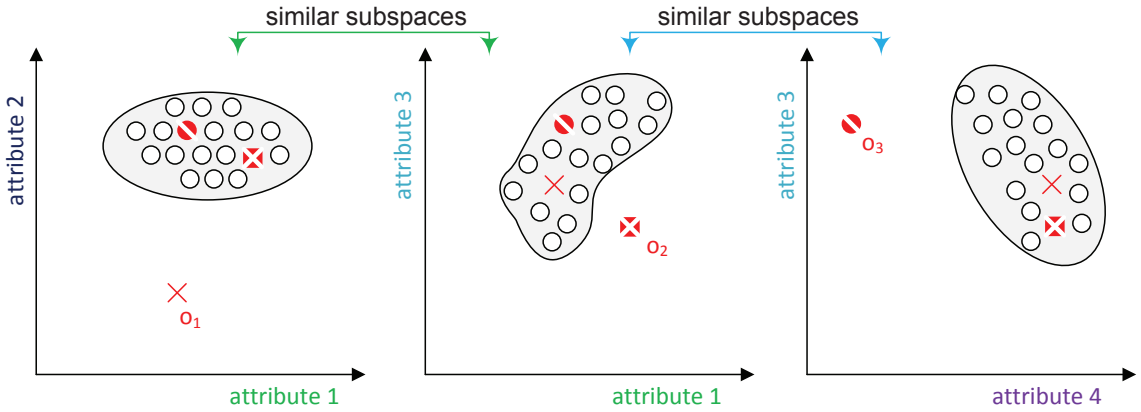


Figure 2. Toy example: Similarity of Subspaces

one subspace cluster, simply assigning a regularity value of 1 to all clustered objects and 0 otherwise, is not feasible.

A second challenge arises from the fact that subspace clusters often constitute redundant patterns, as seconded by a recent evaluation study [10].

Challenge 2: Redundancy of Subspaces
A subspace clustering result $SCR = \{(C_1, S_1), \ldots, (C_k, S_k)\}$ *is usually redundant, i.e., a subspace cluster* (C_i, S_i) *often overlaps (with respect to the clustered objects) with other subspace clusters* (C_j, S_j)*. Typically, these overlaps occur when the subspace projections share many attributes. In the extreme case, a subspace cluster is reflected in all its lower dimensional projections as stated by the following monotonicity property:*

$$(C, S) \in SCR \Rightarrow (C, T) \in SCR \; \forall T \subseteq S$$

Most subspace clustering models obey this monotonicity property [1], [5], [6]. The inverse property is often used to prune subspaces for efficient subspace processing.

As a consequence, each object $o \in (C, S)$ is clustered in all $2^{|S|} - 1$ many lower dimensional subspace projections. Even worse, is the fact that subspace clusters are expected to re-occur in very similar subspaces that share dimensions: $o \in (C, S) \wedge o \in (C, S')$ with $|S \cap S'| \neq 0$. Outlier scores should be aware of the similarity between subspaces, which captures the increasing expectation of shared cluster structures.

The fact that virtually all objects are clustered in multiple views, and, what is more, that these views provide redundant information, is the core challenge for outlier scoring. Still, *OutRank* exploits precisely this fact to derive an evidence measure that takes these multiple views and their information on the varying regularity of objects into account.

Let us illustrate the latter challenge with similar subspaces and our envisioned solution in a toy example. As depicted in Figure 2, we have some 2-dimensional subspaces with three subspace outliers. Each of them is clustered in at least two subspace clusters. Thus, it is hard to distinguish the most deviating outlier. Furthermore, the three subspaces are similar to each other. In particular, the left and the central subspace share "attribute 1", while the central and right one share "attribute 3". Outlier scoring should not be biased by this property as similar clusters are expected, i.e. clustered structures are likely to re-occur in similar subspaces. Handling similarity of subspaces is an open challenge for outlier scoring.

For our scoring functions, we utilize this property of expected clusters in similar subspaces. If one considers the similarity of subspaces we observe an unexpected deviation in o_2. Object o_2 is clustered in two subspaces and deviating in the third (central) one. This is quite unexpected due to the similar subspaces. Objects o_1 and o_3 are not as unexpected. They deviate in dissimilar subspaces. Outlier scoring should account for such expected and unexpected behavior and rank o_2 first.

C. Outlier Scoring Functions

In this section, we introduce three instantiations for the evidence function in Definition 2. As main property, all of these functions can be computed directly out of the subspace clustering result SCR. This keeps the computational overhead for outlier ranking very low as we will show also in our experimental evaluation. Let us start with a baseline scoring function:

Definition 3: Individual Weighting (IW)
Extending Def. 2, we measure the evidence individually for each subspace cluster:

$$score_{IW}(o) = \frac{1}{|SCR|} \cdot \frac{1}{2} \cdot \sum_{\{(C,S) \in SCR \mid o \in C\}} \frac{|C|}{c_{max}} + \frac{|S|}{s_{max}}$$

with $|C|$ *the number of clustered objects, and* $|S|$ *the number of attributes;* c_{max} *the maximal cluster size in* SCR *and* s_{max} *the maximal dimensionality in* SCR*.*

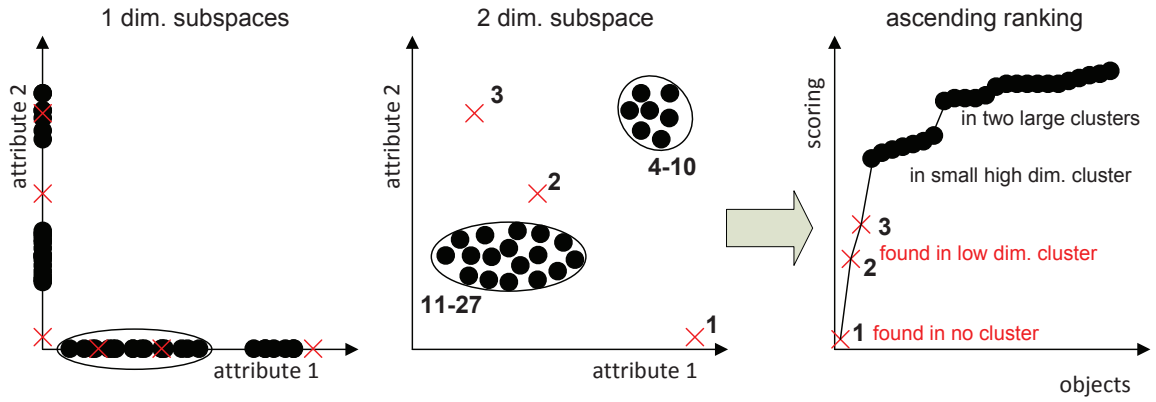


Figure 3. Subspace clusters indicating outliers hidden in subspaces

In our baseline scoring, we simply count the occurrences of an object in a subspace cluster and weight it by the size and dimensionality of (C, S). This reflects the idea that larger subspace clusters with more correlated attributes are stronger evidence for an object's regularity.

Consider a small illustrative example with three outliers in Figure 3. The Figure shows three projections of the data (leftmost figure: two one-dimensional projections to the x-axis and y-axis, respectively, center figure: one two-dimensional projection) and the resulting scoring (rightmost figure). The subspace clusters (circles around areas of high point density) in these low-dimensional projections help identify these outliers which might not be visible in higher dimensional projections. Given the three highlighted subspace clusters we can derive an outlier ranking as depicted in the rightmost figure: objects in no or few and low-dimensional clusters receive low scores, objects which agree with larger and higher dimensional clusters receive higher scores. In this manner, outliers are easily detectable from the ranking as those objects with the lowest scores.

However, this simple measure has clear drawbacks if outliers are not reflected by small and low dimensional clusters. Our first measure does not include a comparison of neither subspaces nor the detected set of clustered objects. As depicted in our previous example (cf. Figure 2), outliers might be detected only due to their unexpected deviation in similar subspaces. Comparing two similar subspaces and the contained clusters leads to a more enhanced scoring. Essentially, redundant clusters do not provide any knowledge for outlier scoring. They simply count each object multiple times and introduce a bias to the overall scoring function. Thus, our more enhanced evidence measures incorporate the similarity of cluster and subspace sets derived from the Jaccard Index: $simObj(C_i, C_j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}$ and $simDim(S_i, S_j) = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$ respectively.

Definition 4:
Comparison based on Subspace Similarity (SS)
For each object o we compare each $(C, S) \in SCR$ with $o \in C$ with all other subspaces $S^* \in SCR$:

$$score_{SS}(o) = \frac{1}{|SCR|} \cdot \sum_{\{(C,S) \in SCR \mid o \in C\}} mean\{subDif(o, S, S^*)\}$$

with $S \neq S^* \in SCR$ and

$$subDif(o, S, S^*) = \begin{cases} 1 - simDim(S, S^*) \\ \quad , \text{if } \forall (C^*, S^*) \in SCR \Rightarrow o \notin C^* \\ 1, \text{ else.} \end{cases}$$

In the extreme case, an object gets the highest $score_{SS}(o) = 1$ if it is clustered in all subspaces. This is the best evidence of being regular. If o is clustered in (C, S) but not in S^* then it depends on the (dis-)similarity of S and S^*. For very similar subspaces one expects that clustered structures reoccur. For redundant subspace clustering models (cf. Challenge 2) this is true due to the monotonicity property. As depicted in Figure 2 it is usually the case that clusters reoccur due to correlated attributes. Definition 4 is aware of this property and expects this situation. In contrast to this expectation, it highlights outliers that show unexpected behavior in such similar subspaces. Lowest scores are assigned to objects $o \in (C, S)$ but not clustered in any of its similar subspaces $simDim(S, S^*) \approx 1$.

Overall, the score aggregates the behavior of o in multiple views, comparing cluster with the residual subspaces in SCR. For each cluster (C, S) we use the harmonic mean of the subspace difference $subDif(o, S, S^*)$ such that strong deviation in one subspace does not dominate the overall score. It enforces low scores only for outliers that show high deviation in many of their similar subspace projections S^*.

In our third scoring function we go even further and consider the possible split of (C, S) in a set of clusters $\{(C_1, S^*), \ldots, (C_j, S^*)\}$ in a similar subspace S^*. A simple example of a split is given in Figure 1: The cluster of white objects in the "social view" is split-up and covered by

two cluster in the "sports view". As given in the following definition, this comparison heavily involves more and more possible reasons for the deviation of the object o:

Definition 5:
Comparison based on Cluster Coverage (CC)
For each object o we compare each $(C,S) \in SCR$ with $o \in C$ with all other subspace cluster sets $\{(C_1,S^*),\ldots,(C_j,S^*)\}$ with high coverage of (C,S):

$$score_{CC}(o) =$$

$$\frac{1}{|SCR|} \cdot \sum_{\{(C,S)\in SCR \mid o\in C\}} mean\{covClust(o,Cov,S^*)\}$$

with $S \neq S^* \in SCR$ and $covClust(o,Cov,S^*) =$

$$\begin{cases} (1-simDim(S,S^*))\cdot \\ \quad mean\{simObj(C,C^*) \mid C^* \in Cov\} \\ \quad \text{, if } \exists (C^*,S^*) \in SCR \wedge o \in C^* \\ \\ simDim(S,S^*)\cdot \\ \quad mean\{(1-simObj(C,C^*) \mid C^* \in Cov\} \\ \quad \text{, else.} \end{cases}$$

, and Cov a set of clusters that covers the objects in C best w.r.t. $simObj(C,C^*)$.

In contrast to the previous definitions, $score_{CC}$ includes the possibility of clusters splitting up in multiple clusters. This can happen as similar subspaces S^* might reveal sub-structures Cov that cover the original subspace cluster (C,S). We utilize the same notion as before and match the "evidence of regularity" to the similarity of subspaces and its contained subspace clusters. In the first case of cluster coverage $covClust(o,Cov,S^*)$ the object is clustered in subspace S^*. Thus, it gets high scores if S and S^* are dissimilar while the detected clusters C and C^* are very similar. This is a good indication for a regular object as it is similarly clustered in different projections. In contrast, the object gets very low scores if it is not clustered in a dissimilar subspace with very similar clusters. The later situation indicates an unexpected outlier which does not follow a similar clustering.

Clearly Definition 5 requires some additional processing in finding the optimal cluster coverage Cov in each subspace. However, it is also the most complex scoring, and we would like to evaluate the quality enhancement by including more and more information. Let us briefly summarize the increase of used information in our three scoring functions:

- baseline scoring (IW): only size and dimensionality of individual clusters are used
- subspace similarity scoring (SS): comparison of multiple subspaces weighted by their similarity
- cluster coverage scoring (CC): comparison of multiple sets of clusters that cover (C,S) weighted by the similarity of subspaces and the similarity of clusters.

D. Discussion of Underlying Subspace Models

Before we study the performance of our scoring functions in an empirical evaluation, let us briefly review and categorize subspace clustering approaches w.r.t. the properties that are relevant for outlier scoring. This should assist in using the appropriate scoring function with the best underlying subspace clustering model. This discussion extends the original publications in subspace clustering w.r.t. their abilities for outlier detection.

Following the terminology of a recent evaluation study [10], we distinguish between four paradigms: (1) *subspace clustering* [4], [5], which shows highly redundant subspace clusters, (2) *projected clustering* [3], [2], which show a partitioning of the data with a binary detection of outliers, (3) *non-redundant subspace clustering* [6], [7], [8], which optimize the result set by removing redundant clusters, and (4) *multiple projected clusters*, i.e. a simple extension of projected clustering based on PROCLUS [2]. Due to space limitations, we cannot review all details of their clustering properties here. For a detailed discussion w.r.t. clustering please refer to the evaluation study [10].

Considering outlier mining, we study properties common to these four clustering paradigms. As in our scoring framework, we abstract from the definition of a cluster (C_i,S_i) in the different models, and resort to the abstract notion of a subspace clustering result $SCR = \{(C_1,S_1),\ldots,(C_k,S_k)\}$.

We base our discussion on a simple measure to distinguish basic properties of different subspace clustering paradigms. An extended empirical study of the performance of different subspace clustering approaches as instantiations to ***OutRank*** is given in the experiments (Section IV). For our formal comparison, we distinguish the overlap of a clustering result:

$$ClusterCount(o,SCR) = |\{(C,S) \in SCR \mid o \in (C,S)\}|$$

$$avgCC(SCR) = \sum_{o\in DB} ClusterCount(o,SCR)/|DB|$$

We distinguish the clustering results by the average number of clusters, in which objects are detected.

- $avgCC(SCR) \geq 2^g$
  with $g = \min\{|S| \mid (C,S) \in SCR\}$
  For all redundant subspace clustering results [4], [5].
- $avgCC(SCR) \leq 1$
  For all partitioning result sets [3], [2].
- $avgCC(SCR) = c$
  with a constant c that can be controlled by the user
  For all non-redundant clustering algorithms optimizing the result set [6], [7], [8].
- $avgCC(SCR) = c$
  with c the number of PROCLUS runs
  For the multiple non-deterministic PROCLUS [2] runs that are combined in one result set.

For our scoring based on cluster coverage (cf. Definition 5) it is crucial to have multiple overlapping subspace clusters. Thus, partitioning approaches with $avgCC(SCR) \leq 1$ should not be used in combination with this score. Due to the high cost of each comparison, $score_{CC}$ should be used with non-redundant subspace clustering [6], [7], [8] or multiple projected clustering results with restricted result size.

For our scoring based on subspace similarity (cf. Definition 4) such restrictions do not apply. It is more flexible and allows also partitioning algorithms [3], [2]. Due to its subspace similarity, $score_{SS}$ is able to distinguish redundant subspaces [4], [5], as well as more enhanced subspace clustering models based on result optimization [6], [7], [8].

IV. Experiments

In the experiments, we study the performance of *OutRank* in comparison with full space outlier ranking methods (LOF [18], and ABOF [19]) and with subspace outlier ranking techniques (OUTRES [13] and SOD [12]). We evaluate our three ranking functions "Individual Weighting" *OutRank(IW)* (cf. Def. 3), "Subspace Similarity" *OutRank(SS)* (cf. Def. 4), and "Cluster Coverage" *OutRank(CC)* (cf. Def. 5). The robustness is studied w.r.t. different subspace clustering models: grid-based *SCHISM* [4]; density-based *INSCY* [6]; approximative *FIRES* [5]; non-redundant *RESCU* [8], as well as two partitioning algorithms *PROCLUS* [2] and *MINECLUS* [3]. And finally, we analyze scalability in terms of database size and dimensionality.

We measure the quality of the obtained outlier rankings using the well-established *ROC* curve, and by calculating the *area under curve* (*AUC*) values [24]. A ROC curve shows the relationship between the false positive rate (x-axis) to the true positive rate (y-axis). The more to the upper left the curve runs, the better its performance. The area under the curve reflects the overall expected performance and provides a way to compare the curves numerically. We ensure comparability of quality results by using publicly available benchmark datasets and re-using synthetic data published by our competitors. For comparability w.r.t. runtime evaluation, we extend the open source framework *SOREX* [25] with our functions, and perform all experiments on a computer cluster, each node equipped with two quad-core Intel Xeon E5540 2.53GHz CPUs running HP XC Linux.

A. Synthetic Data

We start our experimental study by comparing the quality of OutRank with competing algorithms on synthetic data. This benchmark database is used in the evaluation of our most recent competitor OUTRES [13]. It contains 4765 objects, and each object is represented by 16 attributes. In total, 61 outliers deviate from clusters hidden in subspaces with four relevant attributes in each view. Each of the 16 given attributes is used at least once in one of these subspaces. Furthermore, we use a series of synthetic datasets, which are publicly available for subspace cluster evaluation [10]. These datasets are used in order to study the quality and scalability of our ranking functions w.r.t. the different subspace paradigms and increasing number of attributes. In contrast to the first dataset, outliers and clusters are hidden in subspaces with more relevant attributes, increasing with the overall number of given attributes.

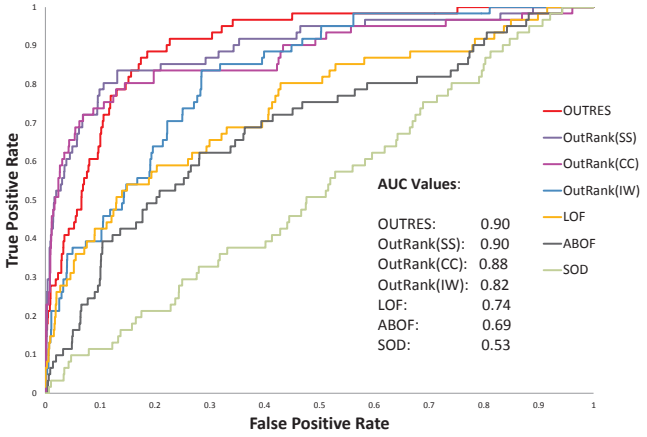
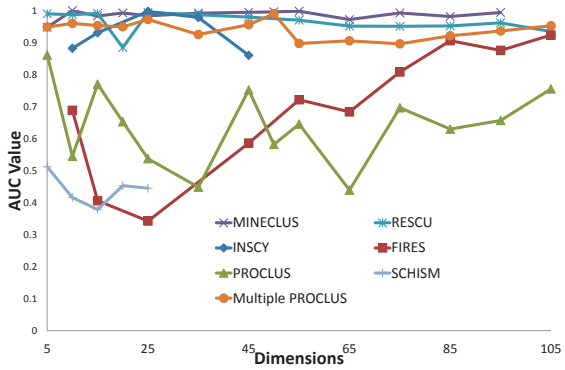


Figure 4. Quality of OutRank vs. competing approaches

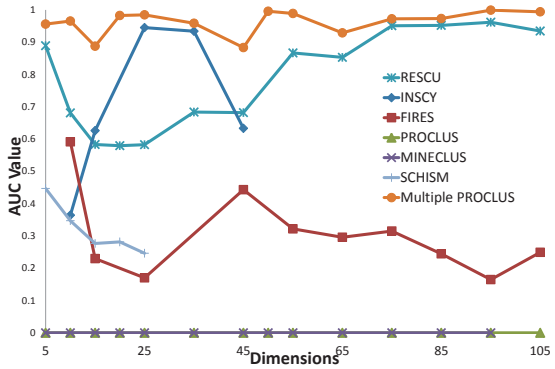
Quality in Comparison to Competitors:
In Figure 4 we show both ROC curves and AUC values for each ranking. Results show that *OutRank* yields best quality: All three scores *(IW)*, *(SS)*, and *(CC)* show better results compared to all full space techniques and also compared to the subspace technique *SOD*. In comparison to *OUTRES*, *OutRank(SS)* and *(CC)* obtain similar AUC values. However, the ROC plot shows the better performance of OutRank. It reaches a higher true positive rate earlier than *OUTRES* and any other competitor.

Quality w.r.t. Subspace Clustering:
As we can see in Figure 5, the performance of OutRank depends on (1) the quality of the given subspace clustering result and (2) the underlying clustering definition. From the clustering perspective, INSCY, RESCU, and MINECLUS have shown the best accuracy in their clustering results [10], [8]. For *OutRank(SS)*, high clustering quality is clearly transfered into a high quality outlier ranking.

The "Cluster Coverage" OutRank(CC) ranking assumes underlying subspace clustering results that provide multiple views, i.e., allow overlap among subspace clusters. Consequently, as seen in Figure 5 (b), partitioning algorithms such as MINECLUS and PROCLUS do not perform well with our "coverage clustering" ranking. Partitioning algorithms assign each object to exactly one subspace cluster, and thus do not provide multiple views. Non-redundant subspace clustering algorithms such as RESCU aim to avoid such overlapping clusters. Thus, they also show slightly worse results for OutRank(CC), but improve with increasing dimensionality



(a) *OutRank(SS)*



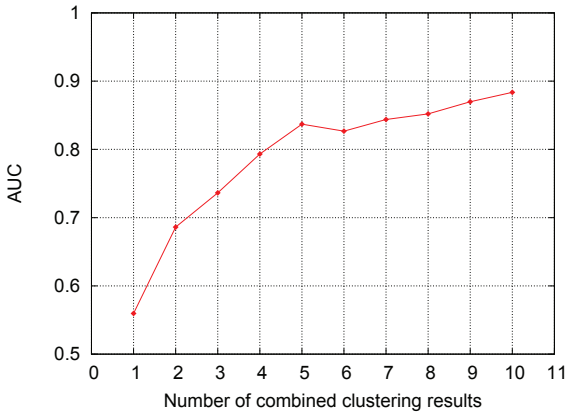
(b) *OutRank(CC)*

Figure 5. Quality w.r.t. different subspace clustering paradigms

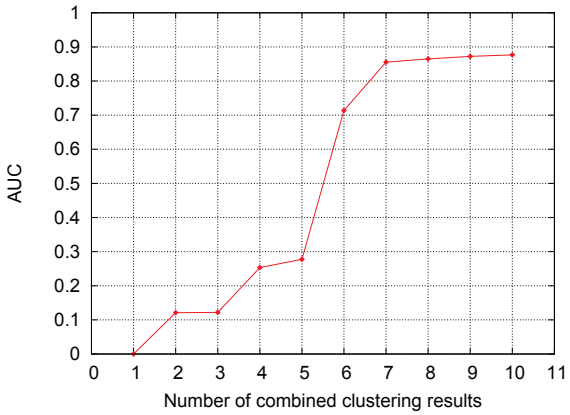
($d \geq 45$) as more and more objects are clustered in multiple views. We demonstrate this effect by including the results of the combination of multiple non-deterministic runs of PROCLUS as *Multiple-PROCLUS*. As we can see, these multiple views greatly improve the performance, and *Multiple-PROCLUS* even shows best performance across all scoring functions and is used as default setting for all other experiments.

We demonstrate this effect of multiple views for *Multiple-PROCLUS* in more detail. Figure 6 shows the quality enhancement with increasing number of views, i.e., number of randomized runs. As we can see, adding views indeed provides more information for the scoring. The higher the number of runs, the higher the number of views of an object, and the higher is the outlier ranking quality. Clearly, OutRank(CC) is more affected by low overlap of subspace clusters.

Runtime Scalability:
Figure 7 shows the runtime scalability for increasing database size and data dimensionality. We compare runtimes of our outlier scoring with runtimes of OUTRES, which showed the best quality among all competitors. For each scoring approach of OutRank, we show the runtimes of the underlying subspace clustering algorithms as well. Our approach outperforms OUTRES as it scales better with increasing number of attributes and also with larger databases.



(a) *OutRank(SS)*



(b) *OutRank(CC)*

Figure 6. Quality w.r.t. multiple views out of several PROCLUS runs

Scalability is achieved by the efficient subspace clustering algorithms [2], [8], which outperform the most recent subspace outlier mining technique [13]. For our baseline function (IW) and for (SS) the overhead for scoring is negligible in comparison with the runtime of the subspace clustering algorithm. On the other hand, (CC) requires higher runtimes if the redundancy of the cluster results is high (e.g., for SCHISM or *Multiple-PROCLUS*). Note, that high quality results with overlapping clusters require such a complex function in order to exploit the cluster properties. Still, the runtime tends to be smaller than the subspace clustering for increasing number of dimensions. Overall, we observe high quality (cf. Fig. 5) and scalable runtime (cf. Fig. 7) of OutRank.

B. Real Data

Finally, we use five real world benchmark datasets from the UCI ML Repository [26]: Diabetes, Breast Cancer Wisconsin (Diagnostic), Ionosphere, Breast Cancer and Arrhythmia with their minority class for outlier evaluation. Due to its stability in quality results on synthetic datasets, we use *Multiple-PROCLUS* with ten runs as default setting for the real world experiments.

dimensions $\lvert\mathcal{D}\rvert$	5	10	20	35	45	55	65	75	85	95	105
OUTRES	2.49	10.69	60.30	331	779.7	-	-	-	-	-	-
PROCLUS *	1.15	1.72	3.16	5.92	8.20	10.19	12.65	15.12	19.75	25.21	28.69
OutRank(IW)	0.03	0.03	0.05	0.07	0.08	0.10	0.12	0.13	0.14	0.16	0.18
OutRank(SS)	0.05	0.08	0.13	0.16	0.18	0.21	0.22	0.22	0.23	0.26	0.26
OutRank(CC)	65.12	79.73	11.14	8.61	7.85	2.05	9.25	9.81	6.82	6.95	15.14
SCHISM	0.02	0.09	21.76	-	-	-	-	-	-	-	-
OutRank(IW)	0.03	0.04	0.05	-	-	-	-	-	-	-	-
OutRank(SS)	0.03	0.04	0.07	-	-	-	-	-	-	-	-
OutRank(CC)	0.10	0.24	0.12	-	-	-	-	-	-	-	-
RESCU	0.11	0.19	4.32	16.69	22.14	49.42	111.8	284.4	414.9	559.3	1275
OutRank(IW)	0.02	0.03	0.05	0.07	0.09	0.10	0.12	0.14	0.15	0.17	0.19
OutRank(SS)	0.03	0.03	0.05	0.07	0.09	0.11	0.13	0.14	0.16	0.18	0.19
OutRank(CC)	0.03	0.04	0.05	0.08	0.09	0.12	0.14	0.15	0.17	0.20	0.21

(a) Runtime (sec.) w.r.t. number of attributes

database size $\lvert DB\rvert$	1595	3722	5848	7975	10102
OUTRES	62.47	105.07	153.42	210.05	268.84
PROCLUS *	2.35	4.81	8.89	12.61	14.71
OutRank(IW)	0.05	0.11	0.17	0.23	0.35
OutRank(SS)	0.13	0.32	0.47	0.67	1.58
OutRank(CC)	8.23	56.59	100.36	79.23	183.52
SCHISM	30.42	72.38	113.57	157.48	119.37
OutRank(IW)	0.04	0.10	0.15	0.21	0.29
OutRank(SS)	0.05	0.11	0.16	0.22	0.32
OutRank(CC)	0.06	0.11	0.17	0.23	0.44
RESCU	5.96	13.85	20.61	32.70	38.14
OutRank(IW)	0.05	0.10	0.16	0.21	0.29
OutRank(SS)	0.05	0.11	0.16	0.22	0.33
OutRank(CC)	0.05	0.11	0.17	0.25	0.40

(b) Runtime (sec.) w.r.t. database size

Figure 7. Runtime scalability

The results of all real world experiments are shown in Fig. 8. Best AUC values are in bold, and high quality results that are within 3% of the best are highlighted as well. OutRank achieves the best results with the highest dimensional datasets. While the traditional full space method LOF shows good performance for small number of attributes, it starts degenerating with increasing dimensionality. OUTRES shows best performance for medium dimensionality. However, it is not able to achieve high quality for Breast, and it does not scale for $d = 128$.

Considering both quality and runtime, OutRank shows best performance with high outlier ranking quality and significantly lower runtime compared to OUTRES. It obviously requires longer runtimes than the full space method LOF since OutRank computes both the subspace clustering and the scoring. On the other hand, we can see that competing full space and subspace approaches do not perform well for increasing number of attributes. By contrast, OutRank can benefit from established subspace clustering methods and achieves best performance on synthetic and real world data.

V. Conclusion

Ranking of outliers is a useful approach to the analysis of deviating objects in the data. Starting from the most unusual objects with respect to patterns in the data, users can study the ranking up to a point where the data appears consistent. Thus, the ranking should correctly reflect the degree of deviation. Traditional approaches fail in uncovering complex deviations hidden in subspace projections of the data. In this work, we address this challenge by *OutRank*, a novel scoring concept based on subspace analysis. Complex deviations are captured by incorporating evidence from subspace clustering results into outlier scores.

Our novel scoring functions capture the evidence as reflected by the main characteristics of the objects w.r.t. subspace clusters. *OutRank* integrates multiple views into the outlier ranking in a principled manner by assessing the information in the entire subspace clustering result. Thus, *OutRank* uncovers outliers that are not detectable in the full attribute space. It outperforms competing approaches from outlier ranking [18], [19], as well as state-of-the-art subspace approaches [13], [12].

VI. Future Work

Due to our abstraction from the underlying method, any (future) subspace clustering algorithm can be utilized for subspace outlier detection. This creates potential for quality

database	d	AUC [%]					Runtime [sec.]				
		LOF	OUTRES	OutRank(IW)	(SS)	(CC)	LOF	OUTRES	OutRank(IW)	(SS)	(CC)
Diabetes	8	**70.98**	58.83	70.07	68.19	68.23	0.3	0.952	1.17	1.19	1.44
Breast (diagnostic)	30	**86.94**	72.97	81.6	77.82	83.72	0.3	10.11	1.3	1.52	203.22
Ionosphere	32	77.97	**83.71**	76.49	73.34	68.67	0.1	227.32	1.71	1.75	1.84
Breast	33	56.42	57.88	**64.75**	60.01	62.2	0.1	107.88	0.66	6.75	0.71
Arrhythmia	128	62.92	-	66.24	65.21	**67.03**	0.5	-	19.35	19.40	13.83

Figure 8. Results on real-world datasets

and efficiency improvements in the future. It could be interesting to study other clustering paradigms that provide multiple clusterings [27]. Recently *alternative clustering*, *disparate clustering*, and *orthogonal clustering* have been proposed [28], [29], [30]. They all provide alternative views on the same data and data space, which could be exploited for outlier scoring as well. These approaches do not work in subspaces, meaning that scoring functions for their notion of multiple views are necessary.

ACKNOWLEDGMENT

This work is supported by the YIG program of KIT as part of the German Excellence Initiative, by a post-doctoral fellowship of the research foundation Flanders (FWO), by the Danish Council for Independent Research - Technology and Production Sciences (FTP) grant 10-081972, and by the German Research Foundation (DFG) within IME Graduate School at KIT.

REFERENCES

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *SIGMOD*, 1998, pp. 94–105.

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

[3] M. L. Yiu and N. Mamoulis, "Frequent-pattern based iterative projected clustering," in *ICDM*, 2003, pp. 689–692.

[4] K. Sequeira and M. Zaki, "SCHISM: A new approach for interesting subspace mining," in *ICDM*, 2004, pp. 186–193.

[5] H.-P. Kriegel, P. Kröger, M. Renz, and S. Wurst, "A generic framework for efficient subspace clustering of high-dimensional data," in *ICDM*, 2005, pp. 250–257.

[6] I. Assent, R. Krieger, E. Müller, and T. Seidl, "INSCY: Indexing subspace clusters with in-process-removal of redundancy," in *ICDM*, 2008, pp. 719–724.

[7] 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*, 2008, pp. 533–541.

[8] E. Müller, I. Assent, S. Günnemann, R. Krieger, and T. Seidl, "Relevant Subspace Clustering: Mining the Most Interesting Non-redundant Concepts in High Dimensional Data," in *ICDM*, 2009, pp. 377–386.

[9] K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong, "A Survey on Enhanced Subspace Clustering," *DMKD*, 2012.

[10] E. Müller, S. Günnemann, I. Assent, and T. Seidl, "Evaluating clustering in subspace projections of high dimensional data," *PVLDB*, vol. 2, no. 1, pp. 1270–1281, 2009.

[11] C. C. Aggarwal and P. S. Yu, "Outlier detection for high dimensional data," in *SIGMOD*, 2001, pp. 37–46.

[12] H.-P. Kriegel, E. Schubert, A. Zimek, and P. Kröger, "Outlier detection in axis-parallel subspaces of high dimensional data," in *PAKDD*, 2009, pp. 831–838.

[13] E. Müller, M. Schiffer, and T. Seidl, "Statistical selection of relevant subspace projections for outlier ranking," in *ICDE*, 2011, pp. 434–445.

[14] E. Müller, I. Assent, U. Steinhausen, and T. Seidl, "OutRank: ranking outliers in high dimensional data," in *ICDE Workshops, DBRank*. IEEE, 2008, pp. 600–603.

[15] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. Wiley, 1987.

[16] V. Chandola, A. Banerjee, and A. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys, Vol. 41, No.3*, July 2009.

[17] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases," in *Proc. KDD*, 1996, pp. 226–231.

[18] M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *SIGMOD*, 2000, pp. 93–104.

[19] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proc. ACM SIGKDD*, 2008.

[20] A. Zimek, E. Schubert, and H.-P. Kriegel, "A survey on unsupervised outlier detection in high-dimensional numerical data," *Statistical Analysis and Data Mining*, 2012.

[21] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos, "LOCI: Fast outlier detection using the local correlation integral," in *ICDE*, 2003, pp. 315–326.

[22] I. Joliffe, *Principal Component Analysis*. Springer, New York, 1986.

[23] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *KDD*, 2005, pp. 157–166.

[24] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[25] E. Müller, M. Schiffer, P. Gerwert, M. Hannen, T. Jansen, and T. Seidl, "SOREX: Subspace outlier ranking exploration toolkit," in *ECML PKDD*, 2010, pp. 607–610.

[26] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html

[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, p. 1220.

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

[29] Z. Qi and I. Davidson, "A principled and flexible framework for finding alternative clusterings," in *KDD*, 2009, pp. 717–726.

[30] D. Niu, J. Dy, and M. Jordan, "Multiple Non-Redundant Spectral Clustering Views," in *ICML*, 2010.