

Enriched Spatial Comparison of Clusterings Through Discovery of Deviating Subspaces

Eric Bae · James Bailey

Received: date / Accepted: date

Abstract Generation and analysis of multiple clusterings is a growing and important research field. A fundamental challenge underpinning this area is how to develop principled methods for assessing and explaining the similarity between two clusterings. A range of clustering similarity indices exist and an important subclass consists of measures for assessing *spatial clustering similarity*. These provide the advantage of being able to take into account properties of the feature space when assessing the similarity of clusterings. However, the output of spatially aware clustering comparison is limited to a single similarity value, which lacks detail for a user. Instead, a user may also wish to understand the degree to which the assessment of clustering similarity is dependent on the choice of feature space.

To this end, we propose a technique for deeper exploration of the spatial similarity between two clusterings. Using as a reference a measure that assesses the spatial similarity of two clusterings in the full feature space, our method discovers deviating subspaces in which the spatial similarity of the two clusterings becomes substantially larger or smaller. Such information provides a starting point for the user to understand the circumstances in which the distance functions associated with each of the two clusterings are behaving similarly or dissimilarly. The core of our method employs a range of pruning techniques to help efficiently enumerate and explore the search space of deviating subspaces. We experimentally assess the effectiveness of our approach using an evaluation with synthetic and real world datasets and demonstrate the potential of our technique for highlighting novel information about spatial similarity between clusterings.

Keywords Clustering similarity · Clustering comparison · Multiple clusterings

Eric Bae
Department of Computing and Information Systems
The University of Melbourne
E-mail: kheb@csse.unimelb.edu.au

James Bailey (corresponding author)
Department of Computing and Information Systems
The University of Melbourne
E-mail: baileyj@unimelb.edu.au

1 Introduction

Clustering is a core technique used in data mining, bioinformatics and pattern recognition and is perhaps the most frequently used method for data exploration. For complex applications and datasets, it is rarely the case that only a single clustering¹ needs to be analyzed. Rather, it is likely that multiple clusterings will need to be considered, in order to gain insight into a domain. Multiple clusterings may arise due to many experiments being carried out at different times, multiple hypotheses being tested, the data containing different perspectives or views, or the existence of legacy, or ‘gold standard’ clusterings being used for external validation.

When exploring a collection of multiple clusterings, a primary task for the user is to be able to assess the similarity between a pair of clusterings. Computation of similarity provides insight for the user into the relationship between two clusterings. This may allow removal of redundant clusterings, selection of interesting clusterings, or increased understanding about clustering evolution. It is also a key step when exploring the convergence properties of a clustering algorithm or assessing the quality of the algorithm’s output when compared to an expert generated clustering.

Many measures have been proposed for measuring the similarity between two clusterings. The largest category of measures are membership based measures, which compare the cluster memberships of objects in the two clusterings. This typically involves either counting the co-occurrence of pairs of objects grouped together in a cluster and not in other clusters, or alternatively using an information theoretic measure to assess the amount of information the object memberships in one clustering provide about the object memberships in the other clustering. Well known examples include the Rand Index (Rand 1971), Jaccard index (Hamers et al. 1989), variation of information (Meila 2007) and normalised mutual information (Strehl and Ghosh 2003). These traditional measures do have some drawbacks, however. They only use information about cluster memberships and are not sensitive to the distances between objects in a cluster. They also cannot be used to compare clusterings of different sets of objects.

To address the limitations with membership based clustering similarity measures, several recent works have proposed techniques for spatially aware clustering comparison (Bae et al. 2010; Coen et al. 2010; Raman et al. 2011). These measures use information about the feature space when assessing distance/similarity between objects, clusters and clusterings. Thus the similarity value that is computed to compare two clusterings depends on the features that have been chosen. This style of approach offers the significant advantage that (unlike membership based measures), these measures are sensitive to distances between objects in a cluster and can be used to compare clusterings performed on different sets of objects.

Whilst using properties of the feature space when comparing clusterings has significant advantages, it can also result in complications. The user may not know which features are most appropriate to use when comparing two clusterings. Furthermore, the output of a spatially aware clustering similarity measure is just a

¹ A clustering is a set of clusters. The clusters are a partition of the set of objects in the dataset.

value between 0 and 1, assessing the degree of similarity. The connection between this similarity value and the chosen feature space may not be obvious to the user.

Given these factors, in this paper we propose a new technique that can be used to enrich the assessment of spatial similarity between two clusterings. Given a reference assessment of clustering similarity in the full feature space, our technique enumerates deviating subspaces in which the clustering similarity is substantially increased or decreased. Such information can provide a starting point for the user to gain insight into the appropriateness and applicability of the distance function being used and the stability of the chosen feature space.

Motivating Scenario: Consider the following hypothetical scenario to motivate our idea. We have two “average” ambulance paramedics, one a novice and the other an expert. Each paramedic is separately shown 100 patient case studies and asked “In your judgement, should this patient be admitted to hospital: {Yes, No, Maybe} ?” The judgements provided by the two paramedics thus correspond to two different clusterings, each having three clusters that partition 100 objects.

A feature space can be constructed to model the information contained in the case studies, using categories of features such as patient information (height, age, weight), vital signs (pulse, respiratory rate, temperature, blood pressure), eye pupil characteristics (dilation, tracking ability), patient speech characteristics (speed, loudness, dynamics) and skin characteristics (blotches, inflammation, oiliness, softness).

The spatial similarity between the two clusterings in this feature space is computed and found to be 0.6, meaning that the clusterings (novice and expert opinions) appear to possess some core similarity. We now wish to identify the subspaces in which the novice and expert judgements appear more similar (their distance functions are closer and more transferable) and the subspaces in which their perspectives appear more dissimilar (their distance functions are less close and not as transferable). For the first task, we find that a subspace consisting of a mixture of vital signs features and speech characteristics yields a clustering similarity of 0.8, which is 33% higher than in the full feature space. This may indicate that the level of training being received by novices for scenarios modelled by this combination of features is adequate and working well. For the second task, we find that a subspace consisting of skin characteristics and eye pupil features yields a clustering similarity of 0.3, which is 50% lower than the similarity in the full feature space. This may indicate that the level of training received by novices for scenarios involving this second combination of features is insufficient and is not allowing them to reach expert standard.

Synthetic Example: In order to visually understand how two clusterings may have higher spatial similarity in a subspace, consider the synthetic example in Figure 1. Here, two clusterings C and C' are being compared in two different feature spaces $A = \{X, Y\}$ (full feature space) and $A' = \{X\}$ (a subspace where $A' \subset A$). Taking into account all the objects and all the clusters, C and C' share more spatial similarity in A' than in A .

We are now in a position to roughly state our problem objective: Given two clusterings, C and C' and feature space A . Let the spatial clustering similarity between C and C' in A be $S(A, C, C')$. Our task is to enumerate all (deviating) subspaces $A' \subset A$ for which i) $\frac{S(A', C, C')}{S(A, C, C')} \geq \delta_1$ (subspaces where there is higher

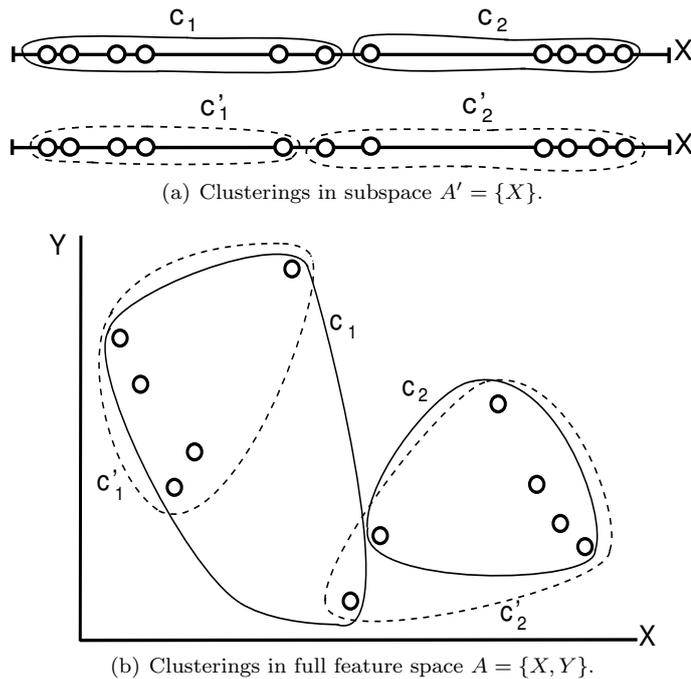


Fig. 1 Comparison of $C = \{c_1, c_2\}$ and $C' = \{c'_1, c'_2\}$ in subspaces $A' = \{X\}$ and $A = \{X, Y\}$. The clusterings have higher spatial similarity in A' than in A . Note that the object-cluster memberships are exactly the same in both A and A' .

similarity) and ii) $\frac{S(A', C, C')}{S(A, C, C')} \leq \delta_2$ (subspaces where there is lower similarity), where $\delta_1 > 1$ and $\delta_2 < 1$ are user specified parameter values.

Contributions: Our first contribution is that we formulate the new problem of deviating subspace discovery for spatial clustering comparison. Our second contribution is that we propose an algorithm (which we call EVE) for enumerating all deviating subspaces. EVE employs, as its base similarity measure, an existing spatially aware similarity method known as *ADCO* (Bae et al. 2010). Efficient enumeration of deviating subspaces is challenging, since the *ADCO* similarity measure is neither monotonic, nor anti-monotonic, nor convertible. This necessitates the use of pruning strategies, based on the determination of upper and lower bounding functions, which themselves have monotonicity properties. We carry out experiments to test the utility of our techniques. We show that i) the notion of deviating subspace identification is meaningful and useful when assessing the similarity between clusterings, ii) the EVE algorithm is able to run effectively on several real world data sets.

2 Related Work

To the best of our knowledge, our problem of discovering deviating subspaces to enrich spatial clustering similarity assessment is novel. An interesting proposal in a similar spirit to our investigation has been developed by Tatti and Vreeken

(2012), where a tiling technique is proposed to assess the similarity between different data mining results (such as clusterings or prediction models). There are also a number of traditional areas of research that are related: i) techniques for clustering similarity measurement, ii) techniques for spatially aware clustering similarity measurement and iii) subspace clustering.

2.1 Current Clustering Comparison Techniques

Roughly speaking, there are three main types of traditional clustering comparison methods.

Pair counting: Methods in this category are based on counting pairs of objects and comparing the agreement and the disagreement between two clusterings. Pairs of objects are classified into four types - N_{11} , N_{10} , N_{01} and N_{00} - where N_{11} is the number of pairs of objects which belong to the same cluster in both clusterings, N_{10} and N_{01} are numbers of pairs which belong to the same cluster in one of the clusterings but not the other, and N_{00} is the number of object pairs belonging to different clusters in both clusterings. N_{11} and N_{00} are treated as agreements and N_{10} and N_{01} are treated as disagreements between the two clusterings. Popular pair counting methods are the Rand index (Rand 1971) and Jaccard index (Hamers et al. 1989) and also the Wallace indices (Wallace 1983) and extensions (Hubert and Arabie 1985).

Set matching: This category of methods is based on measuring the shared set cardinality between two clusterings. The simplest form of set matching technique is called clustering error (Meila 2005), it computes the best matches between clusters (in terms of shared objects) from each of the two clusterings. It returns a value equal to the total number of objects shared between pairs of matched clusters. Other related techniques have also been developed by Larsen and Aone (1999), as well as being generalised to the case of subspace clustering (Günemann et al. 2011).

Information theoretic measures: Examples of these are the normalized mutual information (Strehl and Ghosh 2003) and variation of information (VI) measures (Meila 2007) and adjusted variations (Vinh et al. 2010). These measures utilize the mutual information between two clusterings, which is determined by the conditional probabilities resulting from the number of objects shared between clusters of the two clusterings. The mutual information essentially signifies the amount of information one clustering provides about the other. While NMI normalizes the mutual information with the sum of the two clusterings' entropies, VI uses a different comparison criterion to give the final value.

A limitation with these these methods is that they consider object-to-cluster membership as the primary factor behind clustering comparison. Computing these measures requires no knowledge of underlying feature space being used. This means that they are not sensitive for detecting variation in object distances across clusters. They also cannot be used for comparing clusterings over different collections of objects.

2.2 Spatially-Aware Clustering Comparison

As mentioned earlier, there have been several recent works on spatially-aware clustering similarity measures, which aim to address limitations of the standard membership based measures.

Work by Zhou et al. (2005) proposes a method which takes into account both object memberships and similarity between cluster representatives (cluster centroids). Since it uses information about cluster centroids, we may characterise it as being spatially aware, but it does not directly take account of object-object distances based on the feature space.

Work by Bae et al. (2010) proposes a measure known as *ADCO*, which represents a clustering as a density profile vector and then computes similarity between clusterings using vector operations. We will be using *ADCO* as a base similarity measure throughout the remainder of the paper and exploiting and investigating its properties for subspace enumeration. More detail will be described in Section 3. Compared to the work of Bae et al. (2010), this paper proposes bounding results that are applicable for pruning when employing *ADCO* and also shows how it can be employed for a new application area (deviating subspaces).

Work by Coen et al. (2010) specifies a distance between clusters corresponding to the transportation distance between the collections of objects from each cluster. Based on this distance, it then uses a similarity distance measure to assess the distance between the two clusterings.

An approach by Raman et al. (2011) is based on a Hilbert space representation of clusters. Clusters are modelled as points from a distribution that can be represented as a vector in a reproducing kernel Hilbert space and then compared using a metric on distributions.

2.3 Subspace Clustering

Given that our problem is concerned with both clustering and subspaces, it is natural to consider about connections to the well known area of subspace clustering.

Subspace clustering algorithms such as CLIQUE (Agrawal et al. 1998), MAFIA (Nagesh et al. 1999) and DENCLUE (Hinneburg and Keim 1998) discover clusters (as opposed to clusterings) that are in different subspaces. See the study by Müller et al. (2009) for a critical overview. A subspace cluster is a group of objects and a subset of features. An object may participate in multiple subspace clusters (if not, then it is more properly described as a projected clustering (Aggarwal et al. 1999)).

Recall that our objective is to discover deviating subspaces, subspaces in which the similarity between two clusterings is higher or lower than in the full feature space. Comparing our deviating subspace mining discovery task with subspace clustering, we can make the following observations.

- Both subspace clustering and deviating subspace discovery aim to identify subspaces.
- In deviating subspace discovery, the cluster memberships of objects are fixed ² I.e. objects always retain membership of the same cluster, regardless of whether

² It is reasonable to question why cluster memberships shouldn't change in subspaces. Our setting is that the cluster memberships may have been provided by a human expert and it can

- the full feature space is used, or whether a subspace is used. In subspace clustering, objects may participate in multiple clusters across multiple subspaces.
- In subspace clustering, a subspace usually corresponds to a single cluster. In deviating subspace discovery, each subspace contains a clustering, having the same cluster memberships as the full feature space.
 - The motivation behind deviating subspace discovery is to support similarity comparison of clusterings. It does not aim to discover clusters. The motivation of subspace clustering is to discover high quality clusters hidden in subspaces. Its aim is to discover clusters.

In summary, the focus of deviating subspace discovery is not on producing more accurate clusters by removing features, but rather on analyzing the underlying relationships between two clusterings, within different feature subspaces.

For these reasons, we see deviating subspace discovery as a different task from subspace clustering. Of course there is a possibility that existing algorithms for subspace clustering might somehow be used to assist with deviating subspace discovery. However, it is unclear about how this would be possible. In this paper we adopt a direct approach for enumeration of deviating subspaces, based on properties of a particular clustering similarity function, which is described in the next section.

3 Background on the *ADCO* measure for spatial clustering comparison

Our work is broadly concerned with spatially aware similarity measurement between two clusterings in subspaces. As such, we need to select a spatially aware measure, to be the basis for the technical development of our approach.

We choose to adopt the *ADCO* measure of Bae et al. (2010) as the base measure for our approach. They found it could effectively judge similarity for various types of clusterings. It also has the advantages that it can be used for both continuous and categorical data, it can be used as the basis for alternative clustering generation and it has been found to work efficiently in practice for datasets with many features. Most importantly, for the purposes of this paper, we are able to develop pruning rules based on its properties for efficiently computing its values in subspaces.

At a high level, the *ADCO* measure constructs a spatial histogram for each cluster and represents a clustering as a vector containing the spatial histogram counts for the clusters. The two clusterings can then be compared using vector operations. Intuitively, the output of *ADCO* is a containment judgement between a clustering C and a clustering C' , expressed as “How much of clustering C' is contained in clustering C ?”, or “What percentage of clustering C' is contained in clustering C ?”.

In more detail, we now describe *ADCO*, based on the presentation of Bae et al. (2010). The *ADCO* measure determines the similarity between the two clusterings based on their density profiles along each attribute. Essentially, each attribute’s range is divided into a number of intervals, and the similarity between two clusters corresponds to how closely the object sets from each cluster are distributed

be impossible for the human to manually provide alternative membership judgements for all subspaces.

across these intervals. The similarity between two clusterings then corresponds to the amount of similarity between their component clusters. We begin with some terminology.

Let D be a data set of N objects having R attributes $A = \{a_1, \dots, a_R\}$. Also assume $C = \{c_1, \dots, c_K\}$ and $C' = \{c'_1, \dots, c'_{K'}\}$ are two (hard) clusterings that are to be compared. Each clustering is a partition of the N objects. The *ADCO* similarity value between the two clusterings is denoted as $ADCO(A, C, C')$, where higher values of the measure indicate higher similarity (less dissimilarity). Next we define terms for measuring density.

Definition 1 Given an attribute/feature space $A = \{a_1, a_2, \dots, a_R\}$, let the range of each attribute a_i be divided into Q bins. An **attribute-bin region** is a pair denoted as (i, j) , which corresponds the j -th bin of the i -th attribute. (So there are a total of RQ regions.) The **density of an attribute-bin region** (i, j) is denoted as $dens(i, j)$ and refers to the number of objects in that region expressed as

$$dens(i, j) = |\{d \in D \mid d[a_i] \in (i, j)\}| \quad (1)$$

where $d[a_i]$ is the projection of instance d on attribute a_i . Additionally, the **density of an attribute-bin region for cluster c_k in clustering C** , denoted as $dens_C(k, i, j)$, refers to the number of objects in the region (i, j) , which belong to the cluster c_k of clustering C .

The values of $dens_C(k, i, j)$ for all possible k, i, j form the building blocks of a clustering's 'density profile vector'; in the vector those values are listed in a lexicographical ordering imposed on all attribute-bin regions.

Definition 2 The **density profile** of a clustering C is the following **density profile vector** of C :

$$(dens_C(1, 1, 1), dens_C(1, 1, 2), \dots, dens_C(1, 1, Q), dens_C(1, 2, 1), \dots, dens_C(1, R, Q), dens_C(2, 1, 1), \dots, dens_C(K, R, Q))$$

Suppose C and C' are clusterings with respectively K and K' clusters. We use the following formula on their density profile vectors to determine the degree of similarity between C and C' :

$$sim(A, C, C') = \max_{\rho} \sum_{k=1}^{K_{min}} \sum_{i=1}^R \sum_{j=1}^Q dens_C(k, i, j) \times dens_{C'}(\rho(k), i, j) \quad (2)$$

where ρ ranges over permutations over the cluster IDs of C' and $K_{min} = \min(K, K')$. We note that $sim(A, C, C) = \sum_{k=1}^K \sum_{i=1}^R \sum_{j=1}^Q dens_C(k, i, j)^2$.

By considering all possible permutations ρ , we consider all possible pairings of clusters and select the maximum dot product value corresponding to the best match. The best cluster match may not be the match where the k th cluster in C is matched with the k th cluster in C' . This ensures that the similarity is independent of the assigned cluster labels. To solve the assignment problem of finding the best match between clusters, the Hungarian algorithm (Kuhn 1955) is used, which turns out to operate quite efficiently in practice.

Symbol	Meaning
C, C'	A clustering
A, A', A''	A set of features (a subspace)
A_i^j	Subspace A_i having j number of features
$ADCO(A, C, C')$	Value of $ADCO$ measure comparing C and C' in subspace A (Equation 4)
$LB(A, C, C')$	Lower bound $ADCO$ measure comparing C and C' in subspace A (Equation 15)
$UB(A, C, C')$	Upper bound $ADCO$ measure comparing C and C' in subspace A (Equation 8)
$sim(A, C, C'), sim^{\mathcal{M}}(A, C, C')$	Similarity between C and C' in subspace A choosing maximum permutation (Equation 2)
$sim^{\mathcal{N}}(A, C, C')$	Similarity between C and C' in subspace A choosing minimum permutation (Equation 14)
$NF(A, C, C'), NF^{\mathcal{M}}(A, C, C')$	Normalization factor for C and C' in subspace A choosing maximum (Equation 3)
$NF^{\mathcal{N}}(A, C, C')$	Normalization factor for C and C' in subspace A choosing minimum (Equation 7)

Table 1 Table of Symbols

Lastly, the $ADCO$ measure uses a normalization factor, which corresponds to the maximum achievable similarity when using either of the two clusterings. This is given in equation 3. The $ADCO(A, C, C')$ measure is then shown in equation 4.

$$NF(A, C, C') = \max [sim(A, C, C), sim(A, C', C')] \quad (3)$$

$$ADCO(A, C, C') = \frac{sim(A, C, C')}{NF(A, C, C')} \quad (4)$$

The value of $ADCO$ ranges from 0 to 1, where a lower value indicates higher dissimilarity and a higher value indicates higher similarity.

When clusterings C and C' do not share the same number of clusters, $ADCO$ simply finds the best matching, similar to the clustering error metric described earlier. Note that by varying the Q parameter (the number of bins), one can trade off between the granularity of the density profile and the complexity of computing the $ADCO$ value. Any existing discretization technique can be used for determining bin membership. It has been found that using $Q = 10$ with equi density discretization works well (Bae et al. 2010) and we assume it as a default setting for the remainder of the paper.

A table of symbols that will be used throughout the remainder of the paper is shown in Table 1.

4 The EVE Algorithm for Discovering Deviating Subspaces

We now describe our algorithm for discovering deviating subspaces, which we call *EVE*. It aims to identify subspaces where a pair of clusterings exhibit particularly high or low similarity. By subspaces, we mean any set of attributes which is a subset of the full feature space. This kind of subspace is more properly known as an axis-aligned subspace. E.g. if the full feature space has attributes $\{x, y, z\}$,

then some of the possible subspaces are $\{x, y\}$, $\{x\}$ and $\{y, z\}$. Recalling again our target problem:

Definition 3 *Given two clusterings C and C' of objects described using feature space A , and two user-defined thresholds $\delta_1 > 1$ and $\delta_2 < 1$, the aim of deviating subspace discovery is to enumerate all subspaces A' , where $A' \subset A$, for which either of the following is true*

- *Higher Similarity* : $\frac{ADCO(A', C, C')}{ADCO(A, C, C')} \geq \delta_1$
- *Lower Similarity* : $\frac{ADCO(A', C, C')}{ADCO(A, C, C')} \leq \delta_2$

where $ADCO(A', C, C')$ is the similarity between C and C' in the subspace A' and $ADCO(A, C, C')$ is the similarity between C and C' in the full feature space A .

In order to solve this problem, the EVE algorithm essentially explores the set enumeration tree of all possible subspaces, using a depth first bottom up strategy. In practice, we do not compute subspaces of high and low similarity simultaneously, but rather build a separate enumeration tree to discover each. The $ADCO$ value for each possible subspace is computed at each node in the tree, to determine whether it satisfies the similarity constraint. A challenge though, is that the $ADCO$ function is not well behaved, compared to some well known constraints in data mining, such as frequency. It is not monotonic or anti-monotonic, nor is it even convertible (this last term is defined by Pei et al. (2004)). We assert this in the theorem below.

Theorem 1 *The $ADCO$ similarity function is not*

1. *monotonic*
2. *anti-monotonic*
3. *convertible*.

Consequently, rather more complex pruning is needed in order to make an enumeration algorithm efficient. Otherwise, it would be necessary to enumerate 2^R subspaces, which is clearly infeasible for high dimensional data.

The main idea we use is that for a particular ordering of the enumeration tree, it is possible to bound the $ADCO$ function from above by a monotonically decreasing function, we call UB (upper bound $ADCO$). Hence, once the value of UB drops below $\delta_1 \times ADCO(A, \dots)$ (where A is the full feature space), we know the high similarity constraint will not be satisfied for any superset of the current subspace and we can prune descendants in the tree. We can also bound the $ADCO$ function from below, by a monotonically increasing function that we call LB (lower bound $ADCO$). Hence, once the value of LB rises above $\delta_2 \times ADCO(A, \dots)$, we know the low similarity constraint will not be satisfied for any superset of the current subspace and we can prune descendants in the tree.

4.1 Calculating the $ADCO$ Measure When a Subspace is Grown

For calculating the $ADCO$ value between two clusterings in any subspace $A' = \{a_1, a_2, \dots, a_{R'}\}$ of A , we can simply use Equation 4. The value of $sim^M(A', C, C')$ is computed by replacing A by A' in Equation 4 and consequently computing a

density profile that has R' attributes rather than R in Equation 2. Note that we henceforth use $\text{sim}^{\mathcal{M}}(A', C, C')$ instead of $\text{sim}(A', C, C')$ to emphasise that it is the maximum scalar product. This is necessary as we also introduce the minimum scalar product $\text{sim}^{\mathcal{N}}(A', C, C')$ and will later need to distinguish them.

This substitution is possible, since the density information of each attribute is independently determined. In EVE, subspaces are grown (extended) one attribute at a time, during the enumeration process. Therefore, let us consider calculating the $ADCO$ value when a subspace is extended.

Assume subspace $A' = \{a'_1, a'_2, \dots, a'_{R'}\}$ is being extended by the singleton subspace $A'' = \{a''_1\}$, where $A' \subset A, A'' \subset A$ and $A' \cap A'' = \{\}$. Let the merged subspace be $A' \cup A''$. Similarity in the new merged subspace can be expressed as follows:

$$ADCO(A' \cup A'', C, C') = \frac{\text{sim}^{\mathcal{M}}(A' \cup A'', C, C')}{NF^{\mathcal{M}}(A' \cup A'', C, C')} \quad (5)$$

$$= \frac{\text{sim}^{\mathcal{P}}(A', C, C') + \text{sim}^{\mathcal{P}}(A'', C, C')}{NF^{\mathcal{C}}(A', C, C') + NF^{\mathcal{C}}(A'', C, C')} \quad (6)$$

where $\text{sim}^{\mathcal{P}}(A', C, C')$ refers to computing the similarity using some particular permutation \mathcal{P} and $NF^{\mathcal{C}}(A', C, C')$ refers to computing the normalization factor choosing (say) clustering \mathcal{C} as the maximum of the two clusterings. Observe that the second equality is sustained when the same permutation \mathcal{P} is used in both $\text{sim}^{\mathcal{P}}(A', C, C')$ and $\text{sim}^{\mathcal{P}}(A'', C, C')$, to give the maximum similarity value $\text{sim}^{\mathcal{M}}(A' \cup A'', C, C')$.

Furthermore, the same clustering \mathcal{C} must be chosen in both $NF^{\mathcal{C}}(A', C, C')$ and $NF^{\mathcal{C}}(A'', C, C')$ when determining the maximum normalization factor $NF^{\mathcal{M}}(A' \cup A'', C, C')$. Similar to $\text{sim}^{\mathcal{M}}(A', C, C')$, we will henceforth denote the normalization factor shown in Equation 3 here as $NF^{\mathcal{M}}(A' \cup A'', C, C')$ to explicitly indicate that it is the maximum normalization factor. We will also use $NF^{\mathcal{N}}(A', C, C')$ which selects the minimum normalizing factor. This highlights the point that the choices which would maximize the $ADCO$ value for individual subspaces may not be the choices that achieve the maximum $ADCO$ value for a larger subspace in which they are contained.

4.2 An Upper Bound and a Lower Bound for the $ADCO$ Value in Any Subspace

We are going to bound the $ADCO$ function from above by a function UB , which takes its maximum value for some singleton subspace and thereafter decreases monotonically, provided subspaces are grown by following certain ordering conditions. Similarly, we are going to bound the $ADCO$ function below by a function LB , which takes its minimum value for some singleton subspace and thereafter increases monotonically, provided subspaces are grown following certain ordering conditions. We consider the upper bound case first:

4.2.1 An Upper Bound

We begin with a definition for the UB function:

$$NF^{\mathcal{N}}(A, C, C') = \min \left[sim^{\mathcal{M}}(A, C, C), sim^{\mathcal{M}}(A, C', C') \right] \quad (7)$$

$$UB(A, C, C') = \frac{sim^{\mathcal{M}}(A, C, C')}{NF^{\mathcal{N}}(A, C, C')} \quad (8)$$

where in Equation 7, the clustering that yields the minimum value for the normalization factor is chosen (as opposed to Equation 3, which selects the maximum), while in Equation 8, UB is calculated with $NF^{\mathcal{N}}(A, C, C')$ (instead of $NF^{\mathcal{M}}(A, C, C')$ as in Equation 4). The following properties are then straightforward to check for any $A' \subseteq A$:

$$NF^{\mathcal{N}}(A', C, C') \leq NF^{\mathcal{M}}(A', C, C') \quad (9)$$

$$ADCO(A', C, C') \leq UB(A', C, C'). \quad (10)$$

We will now show that UB takes its maximum value for a subspace with just a single attribute (we call this a *base subspace*). Let A_i be a subspace and let A_i^j indicate that subspace A_i has dimensionality j .

Lemma 1 *Consider the set of UB values for all base subspaces (1-dimensional subspaces):*

$$\left\{ UB(A_1^1, C, C'), UB(A_2^1, C, C'), \dots, UB(A_R^1, C, C') \right\}$$

Let $UB(A_{\mathcal{M}}^1, C, C')$ be the maximum value in this set.

Then $UB(A_{\mathcal{M}}^1, C, C') \geq UB(A', C, C')$ where A' is any subspace with an arbitrary number of attributes and $A' \subseteq A$.

The above lemma states that there exists a base subspace $A_{\mathcal{M}}^1$, which has the highest UB value compared to any other base subspaces. Moreover, it has a greater than or equal to UB value compared to any R' -dimensional subspaces A' , where $R' \leq R$. The value of $UB(A_{\mathcal{M}}^1, C, C')$ is, therefore, the upper bound for UB values of all subspaces in A . Proving that $UB(A_{\mathcal{M}}^1, C, C')$ is the highest value amongst all 1-dimensional subspaces is trivial. We now provide the proof.

Proof $UB(A_{\mathcal{M}}^1, C, C')$ satisfies :

$$UB(A_{\mathcal{M}}^1, C, C') = \max\{UB(A_1^1, C, C'), \dots, UB(A_R^1, C, C')\}$$

Now, let A' be a R' -dimensional subspace containing the subspace $A_{\mathcal{M}}^1$. Using Equation 5 and the property defined in Equation 9, we can re-write $UB(A', C, C')$ as follows:

$$UB(A', C, C') = \frac{sim^{\mathcal{M}}(A', C, C')}{NF^{\mathcal{N}}(A', C, C')} = \frac{sim^{\mathcal{P}}(A_1^1, C, C') + \dots + sim^{\mathcal{P}}(A_{\mathcal{M}}^1, C, C') + \dots + sim^{\mathcal{P}}(A_{R'}^1, C, C')}{NF^{\mathcal{C}}(A_1^1, C, C') + \dots + NF^{\mathcal{C}}(A_{\mathcal{M}}^1, C, C') + \dots + NF^{\mathcal{C}}(A_{R'}^1, C, C')} \quad (11)$$

and given the above, the following is also true :

$$\frac{\text{sim}^{\mathcal{M}}(A', C, C')}{\text{NF}^{\mathcal{N}}(A', C, C')} \leq \max \left[\frac{\text{sim}^{\mathcal{M}}(A_1^1, C, C')}{\text{NF}^{\mathcal{N}}(A_1^1, C, C')}, \dots, \frac{\text{sim}^{\mathcal{M}}(A_{\mathcal{M}}^1, C, C')}{\text{NF}^{\mathcal{N}}(A_{\mathcal{M}}^1, C, C')}, \dots, \frac{\text{sim}^{\mathcal{M}}(A_{R'}^1, C, C')}{\text{NF}^{\mathcal{N}}(A_{R'}^1, C, C')} \right] \quad (12)$$

and so, the theorem holds.

To see why Equation 12 is true, let us rewrite it as $\frac{A+C+\dots+X}{B+D+\dots+Y} \leq \max \left[\frac{A}{B}, \frac{C}{D}, \dots, \frac{X}{Y} \right]$. Let us assume that $\max \left[\frac{A}{B}, \frac{C}{D}, \dots, \frac{X}{Y} \right] = \frac{A}{B}$. Then we have

$$\frac{A+C+\dots+X}{B+D+\dots+Y} \leq \frac{A}{B},$$

$$AB + BC + \dots + BX \leq AB + AD + \dots + AY \quad (13)$$

and since $\frac{A}{B} \geq \frac{C}{D}$ therefore $AD \geq BC$ and subsequently since $\frac{A}{B} \geq \frac{X}{Y}$, then $AY \geq BX$ and thus, after these manipulations we can conclude that Equation 13 is true and thus Equation 12 is true. \square

From the above proof, we can state that there is a base subspace $A_{\mathcal{M}}^1$ with the highest UB value compared to all other subspaces. Therefore, in the enumeration process of EVE, we are certain that the $UB(A_{\mathcal{M}}^1, C, C') \geq UB(A', C, C')$ for any subspace A' in A . Furthermore, by following the property in Equation 10, $UB(A_{\mathcal{M}}^1, C, C') \geq ADCO(A', C, C')$. Hence we can begin our branch in the enumeration tree using $UB(A_{\mathcal{M}}^1, C, C')$ at the top.

4.2.2 A Lower Bound

The lower bound can be defined in a similar manner to the upper bound. We begin with a definition for the LB function:

$$\text{sim}^{\mathcal{N}}(A, C, C') = \min_{\rho} \sum_{k=1}^K \sum_{i=1}^R \sum_{j=1}^Q \text{dens}_{c_k}(a_i, b_j) \cdot \text{dens}_{c'_k}(a_i, b_j) \quad (14)$$

$$LB(A, C, C') = \frac{\text{sim}^{\mathcal{N}}(A, C, C')}{\text{NF}^{\mathcal{M}}(A, C, C')}. \quad (15)$$

Equation 14 selects the permutation that gives the *minimum* scalar product value instead of the maximum value that was selected by Equation 2. In Equation 15, LB is computed using $\text{sim}^{\mathcal{N}}$. The following properties are then straightforward to check for any $A' \subseteq A$:

$$\text{sim}^{\mathcal{N}}(A', C, C') \leq \text{sim}^{\mathcal{M}}(A', C, C') \quad (16)$$

$$LB(A', C, C') \leq ADCO(A', C, C'). \quad (17)$$

We will now show that LB takes its minimum value for some base subspace.

Lemma 2 Consider the set of LB values for all base subspaces :

$$\left\{ LB(A_1^1, C, C'), LB(A_2^1, C, C'), \dots, LB(A_R^1, C, C') \right\}.$$

Let $LB(A_{\mathcal{N}}^1, C, C')$ be the minimum value in this set. Then, $LB(A_{\mathcal{N}}^1, C, C') \leq LB(A', C, C')$ where A' is any subspace with arbitrary number of attributes and $A' \subseteq A$.

The above lemma states that there exists a base subspace $A_{\mathcal{N}}^1$ which has the lowest LB value compared to all other base subspaces and moreover has value less than or equal to the LB value of any other higher dimensional subspaces of A . Therefore, $LB_{A_{\mathcal{N}}^1}$ is the lower bound for LB values of all subspaces of A .

Proof The proof for this is symmetric to that of Lemma 1. \square

From the above proof, we can assert that there is a 1-dimensional subspace $A_{\mathcal{N}}^1$ having the lowest LB value compared with all other subspaces. Therefore, in the enumeration process of EVE, we can be certain that $LB(A_{\mathcal{N}}^1, C, C') \leq LB(A', C, C')$ for any subspace A' .

4.3 Greedy Prefix Monotonicity and Anti-monotonicity

Having established upper and lower bounds we are now interested in how these values may change as bottom-up subspace enumeration proceeds.

Specifically, our goal is to find an *ordering of the base subspaces* such that, when growth in the enumeration tree follows this order, UB and LB are monotonically decreasing and increasing respectively. In other words, we will establish these functions which can obey a type of *prefix (anti-)monotonic property* (Pei et al. 2002).

Rather interestingly, we are unable to determine this prefix order in a static fashion. Instead, we can only uncover it as the enumeration tree is explored. Recall that a function f is monotonically increasing if whenever $A \subseteq B$, then $f(A) \leq f(B)$. f is monotonically decreasing if whenever $A \subseteq B$, then $f(A) \geq f(B)$.

4.3.1 Monotonically Decreasing UB Values

Suppose we begin with the base subspace $A_{\mathcal{M}}^1$ and are now moving down the leftmost branch of the enumeration tree, to grow a 2-dimensional subspace. Which base subspace should be merged next with $A_{\mathcal{M}}^1$ in order that UB decreases ?

The answer is that we should grow using the base subspace $A_{\mathcal{O}}^1$, such that there is a minimum decrease in similarity value. That is, the value of $UB(A_{\mathcal{M}}^1, C, C') - UB(A_{\mathcal{O}}^1 \cup A_{\mathcal{M}}^1, C, C')$ is minimized. In essence, this can be regarded as a “greedy” type of growth.

More generally, suppose we are currently at some node in the enumeration tree corresponding to subspace S^q and that any of the attributes (base subspaces) $\{a_1, \dots, a_k\}$ could be used for forming a subspace of size $q + 1$. Then we should choose to grow using the attribute a_i which minimizes the value of $UB(S^q, C, C') -$

$UB(S^q \cup \{a_i\}, C, C')$. We will call this type of ordering strategy a *greedy prefix ordering*.

Thus, at each node in the tree, our ordering must be *dynamically chosen*, by testing all possible extensions and greedily choosing the one for which the change in UB value is minimum. Using this ordering strategy, it is guaranteed that the resulting UB function will be monotonically decreasing, when moving down along each branch of the enumeration tree. This is captured in the following theorem.

Theorem 2 *Let $S = \{a_1, \dots, a_k\}$ be a subspace, whose attributes are ordered according to the greedy prefix ordering method described above. If subspace p is any prefix of subspace S (according to the enumeration tree order), then*

1. $UB(p, C, C') \geq UB(S, C, C')$ and
2. $UB(p, C, C') \geq UB(S', C, C')$, for any subspace S' for which $|S'| = |p|$ and $S' \subseteq \{a_1, \dots, a_k\}$.

Proof: We show the proof for the above theorem by induction on the size k of the subspace. For $k = 2$, we have seen from lemma 1, that there exists some base subspace whose UB value is greater than that for any superspace in the enumeration tree. This base subspace will correspond to $A_{\mathcal{M}}^1 = \{a_1\}$. It therefore, follows that the following is true:

$$UB(\{a_1\}, C, C') \geq UB(\{a_1, a_2\}, C, C') \quad (18)$$

and it is true that

$$UB(\{a_1, a_2\}, C, C') \geq UB(\{a_2, a_1\}, C, C') \quad (19)$$

For the induction step, assume the theorem is true for all subspaces of size less than or equal to k , we need to show it is true for subspaces of size $k + 1$. To show it is true for $k + 1$, we will need to establish that

$$UB(\{a_1, \dots, a_{k+1}\}, C, C') \leq UB(\{a_1, \dots, a_k\}, C, C'). \quad (20)$$

We can show that the following statement is true

$$UB(\{a_1, \dots, a_{k+1}\}, C, C') \leq \max[UB(\{a_1, \dots, a_k\}, C, C'), UB(\{a_2, \dots, a_{k+1}\}, C, C')] \quad (21)$$

if by the induction step the following is true

$$UB(\{a_2, \dots, a_{k+1}\}, C, C') \leq UB(\{a_1, \dots, a_k\}, C, C'). \quad (22)$$

The above condition is directly what Equation 19 states, that $UB(p, C, C') \geq UB(S', C, C')$ where $|S'| = |p|$ and $S' \subseteq \{a_1, \dots, a_k\}$.

To show this consider the case when 2-dimensional subspaces are being formed. Let $A_{\mathcal{M}}^1$ be the subspace with the maximum UB value and it is merged with $A_{\mathcal{O}}^1$ to create $A_{\{\mathcal{O}, \mathcal{M}\}}^2$. Given two other base subspaces $A_{\mathcal{U}}^1$ and $A_{\mathcal{V}}^1$, we show the following is true :

$$\begin{aligned}
UB(A_{\mathcal{M}}^1, C, C') - UB(A_{\{O, \mathcal{M}\}}^2, C, C') &\leq \\
UB(A_{\mathcal{M}}^1, C, C') - UB(A_{\{U, \mathcal{M}\}}^2, C, C') &\leq \\
UB(A_{\mathcal{M}}^1, C, C') - UB(A_{\{V, \mathcal{M}\}}^2, C, C'). &
\end{aligned} \tag{23}$$

For brevity, let us represent the above equation as follows:

$$\left(\frac{A}{B} - \frac{A+C}{B+D} \right) \leq \left(\frac{A}{B} - \frac{A+X}{B+Y} \right) \leq \left(\frac{A}{B} - \frac{A+X'}{B+Y'} \right) \tag{24}$$

where A, C, X and X' corresponds to $sim^{\mathcal{M}}(A_{\mathcal{M}}^1, ..)$, $sim^{\mathcal{M}}(A_{\mathcal{O}}^1, ..)$, $sim^{\mathcal{M}}(A_{\mathcal{U}}^1, ..)$ and $sim^{\mathcal{M}}(A_{\mathcal{V}}^1, ..)$ respectively. The variables B, D, Y and Y' corresponds to $NF^{\mathcal{N}}(A_{\mathcal{M}}^1, ..)$, $NF^{\mathcal{N}}(A_{\mathcal{O}}^1, ..)$, $NF^{\mathcal{N}}(A_{\mathcal{U}}^1, ..)$ and $NF^{\mathcal{N}}(A_{\mathcal{V}}^1, ..)$ respectively.

From the above, the following properties are deduced :

$$\begin{aligned}
\frac{A}{B} &\geq \frac{C}{D}, \quad AD \geq BC, \quad \frac{A}{B} \geq \frac{X}{Y}, \quad AY \geq BX, \\
\frac{A}{B} &\geq \frac{X'}{Y'}, \quad AY' \geq BX', \quad \frac{C}{D} \geq \frac{X}{Y}, \quad CY \geq DX, \\
\frac{C}{D} &\geq \frac{X'}{Y'}, \quad CY' \geq DX', \quad \frac{X}{Y} \geq \frac{X'}{Y'}, \quad XY' \geq X'Y.
\end{aligned} \tag{25}$$

With these conditions, let us now prove the following :

$$\left(\frac{A}{B} - \frac{A+C}{B+D} \right) \leq \left(\frac{A}{B} - \frac{A+X'}{B+Y'} \right) \tag{26}$$

by re-writing it as follows:

$$\begin{aligned}
\frac{A+X}{B+Y} &\geq \frac{X+X'}{Y+Y'}, \\
AY + AY' + XY + XY' &\geq BX + BX' + XY + X'Y
\end{aligned} \tag{27}$$

and by using the properties defined in Equation 25, the above equation yields a positive value. This ensures that selecting $A_{\mathcal{O}}^1$ and merging with $A_{\mathcal{M}}^1$ returns the highest $UB(A_{\mathcal{M}}^2, C, C')$ value in the 2-dimensional subspace. Moreover, this property suggests $A_{\mathcal{O}}^1$ succeeds $A_{\mathcal{M}}^1$ in the ordered set of base subspaces prior to proceeding the enumeration steps.

Consider now a general case of showing that Equation 22 is true for a subspace A' of size k , where $R \geq k > 2$. Following Equation 23, we can express this as

$$\begin{aligned}
UB(\{a_1, \dots, a_{k-1}\}, C, C') - UB(\{a_1, \dots, a_k\}, C, C') &\leq \\
UB(\{a_1, \dots, a_{k-1}\}, C, C') - UB(\{a_2, \dots, a_{k+1}\}, C, C') &\leq \\
UB(\{a_1, \dots, a_{k-1}\}, C, C') - UB(\{a_3, \dots, a_{k+2}\}, C, C') &
\end{aligned} \tag{28}$$

For brevity, let us first represent $UB(\{a_1, \dots, a_{k-1}\}, C, C')$ as follows:

$$UB(\{a_1, \dots, a_{k-1}\}, C, C') = \underbrace{\frac{A_1 + A_2 + \dots + A_{k-1}}{B_1 + B_2 + \dots + B_{k-1}}}_{\mathbb{B}}^{\mathbb{A}} \quad (29)$$

where we label the terms of numerator as \mathbb{A} and the denominators as \mathbb{B} . Equation 28 can now be expressed as

$$\left(\frac{\mathbb{A}}{\mathbb{B}} - \frac{\mathbb{A} + A_k}{\mathbb{B} + B_k} \right) \leq \left(\frac{\mathbb{A}}{\mathbb{B}} - \frac{\mathbb{A} + A_{k+1}}{\mathbb{B} + B_{k+1}} \right) \leq \left(\frac{\mathbb{A}}{\mathbb{B}} - \frac{\mathbb{A} + A_{k+2}}{\mathbb{B} + B_{k+2}} \right). \quad (30)$$

From Equation 30, we can deduce the properties similar to Equation 25 as follows:

$$\begin{aligned} \frac{\mathbb{A}}{\mathbb{B}} &\geq \frac{A_k}{B_k}, \quad \mathbb{A}B_k \geq \mathbb{B}A_k, \quad \frac{\mathbb{A}}{\mathbb{B}} \geq \frac{A_{k+1}}{B_{k+1}}, \quad \mathbb{A}B_{k+1} \geq \mathbb{B}A_{k+1} \\ \frac{\mathbb{A}}{\mathbb{B}} &\geq \frac{A_{k+2}}{B_{k+2}}, \quad \mathbb{A}B_{k+2} \geq \mathbb{B}A_{k+2}, \quad \frac{A_k}{B_k} \geq \frac{A_{k+2}}{B_{k+2}}, \quad A_k B_{k+2} \geq B_k A_{k+2}, \\ \frac{A_k}{B_k} &\geq \frac{A_{k+2}}{B_{k+2}}, \quad A_k B_{k+2} \geq B_k A_{k+2}, \quad \frac{A_{k+1}}{B_{k+1}} \geq \frac{A_{k+2}}{B_{k+2}}, \quad A_{k+1} B_{k+2} \geq B_{k+1} A_{k+2}, \end{aligned} \quad (31)$$

and to prove $\left(\frac{\mathbb{A}}{\mathbb{B}} - \frac{\mathbb{A} + A_k}{\mathbb{B} + B_k} \right) \leq \left(\frac{\mathbb{A}}{\mathbb{B}} - \frac{\mathbb{A} + A_{k+2}}{\mathbb{B} + B_{k+2}} \right)$, we can re-write it as

$$\begin{aligned} \frac{\mathbb{A} + A_k}{\mathbb{B} + B_k} &\geq \frac{A_k + A_{k+2}}{B_k + B_{k+2}}, \\ \mathbb{A}B_k + \mathbb{A}B_{k+2} + A_k B_k + A_k B_{k+2} &\geq \mathbb{B}A_k + \mathbb{B}A_{k+2} + A_k B_k + A_{k+2} B_k \end{aligned} \quad (32)$$

and by using the properties defined in Equation 31, Equation 32 yields a positive value and proves Equation 22. \square

4.3.2 Monotonically Increasing LB Values

From the previous section, we know that the smallest LB value will occur for a base subspace and is in fact $A_{\mathcal{N}}^1$. Our subspace ordering approach will be symmetric to the monotonically decreasing case, again using a greedy prefix ordering.

Suppose we are currently at some node in the enumeration tree corresponding to subspace S^q and that any of the attributes (base subspaces) $\{a_1, \dots, a_k\}$ could be used for forming a subspace of size $q+1$. Then we should choose to grow using the attribute a_i which minimizes the value of

$$LB(S_q \cup \{a_i\}, C, C') - LB(S_q, C, C').$$

So again, our ordering must be dynamically determined, by testing all possible extensions and greedily choosing the one for the change is minimum. Using this ordering strategy, it is guaranteed that the resulting LB function must be monotonically increasing, moving down along each branch of the enumeration tree. This is stated in the following theorem.

Theorem 3 Let $S = \{a_1, \dots, a_k\}$ be a subspace, whose attributes are ordered according to the greedy prefix ordering method described above. If subspace p is any prefix of subspace S (according to the enumeration tree order), then

1. $LB(p, C, C') \leq LB(S, C, C')$ and
2. $LB(p, C, C') \leq LB(S', C, C')$, for any subspace S' for which $|S'| = |p|$ and $S' \subseteq \{a_1, \dots, a_k\}$.

Proof : The proof for monotonically increasing LB values is symmetric to that of Theorem 2.

4.4 Incrementally Reusing $ADCO$ Values of Base Subspaces

We now briefly discuss where it is possible to use incremental strategies when exploring the search space. Consider a subspace $A' = \{a_1, a_2, \dots, a_{R'}\}$. We know that Equation 2 requires all values from the permutation function ρ to be computed before the maximum value can be selected. Since each attribute independently contributes to the overall similarity, we can store values of the scalar product from all permutations for each base subspace $\{a_i\}$ in an initial processing phase. These values can then be reused as required for a merged subspace A' , to calculate $sim^{\mathcal{M}}(A', C, C')$ as below :

$$sim^{\mathcal{M}}(A', C, C') = \max \sum_{i=1}^{|\rho|} \sum_{j=1}^{R'} sim^i(j, C, C')$$

where $|\rho|$ is the number of possible different permutations required and $\sum_{j=1}^{R'} sim^{\rho}(j, C_1, C_2)$ is the stored similarity value for the j -th attribute for permutation i . Similarly, for each base subspace a_i , $sim^{\mathcal{M}}(a_i, C_1, C_1)$ and $sim^{\mathcal{M}}(a_i, C_2, C_2)$ need to be calculated to select the normalization factor $NF^{\mathcal{M}}(a_i, C, C')$. These values can also be stored and reused for evaluating the new normalization factor for A' as below :

$$NF^{\mathcal{M}}(A', C, C') = \max \left[\sum_{j=1}^{R'} sim^{\mathcal{M}}(a_j, C, C), \sum_{j=1}^{R'} sim^{\mathcal{M}}(a_j, C', C') \right].$$

The stored values can also be used to calculate the values of $NF^{\mathcal{N}}(A', C, C')$, $sim^{\mathcal{N}}(A', C, C')$, $LB(A', C, C')$ and $UB(A', C, C')$ as needed. Lastly, the UB and LB values required for the greedy choice of subspace extension, can be reused once that part of the enumeration tree begins to be processed.

4.5 Algorithm Description

We present the pseudo code for EVE in which subspaces of high similarity and dissimilarity between clusterings are found respectively in Algorithm 1 and Algorithm 2.

Algorithm 1 EVE algorithm for finding subspaces of high similarity between C and C'

Require: clusterings C and C' , full feature space A , threshold $\delta_1 > 1$

Ensure: all subspaces $A' \subset A$ in which the similarity between C and C' satisfy the constraint δ_1 are returned

```

1: for  $i = 1$  to  $R$  do
2:   Calculate  $ADCO$  for all base subspaces
3:   Calculate  $UB$  for all base subspaces
4:    $SortedBase = \text{Sort values of } UB \text{ for the base subspaces}$ 
5: end for
6: for  $i = 1$  to  $R$  do
7:    $subspace = \{a_i\}$  where  $a_i \in SortedBase$  ( $a_i$  is the first element in the sorted list (has the largest  $UB$  value))
8:   if  $UB(subspace, C, C') \geq \delta_1 \times ADCO(A, C, C')$  then
9:     if  $ADCO(subspace, C, C') \geq \delta_1 \times ADCO(A, C, C')$  then
10:      output  $subspace$ 
11:     end if
12:     Greedily select next attribute  $a_j$ 
13:     repeat
14:        $subspace = subspace \cup \{a_j\}$ 
15:       if  $UB(subspace, C, C') \geq \delta_1 \times ADCO(A, C, C')$  then
16:         if  $ADCO(subspace, C, C') \geq \delta_1 \times ADCO(A, C, C')$  then
17:           output  $subspace$ 
18:         end if
19:       else
20:          $subspace = subspace \setminus \{a_j\}$ 
21:         Backtrack to next node in enumeration tree
22:       end if
23:       Greedily select next attribute  $a_j$ 
24:     until no attributes remain to be added
25:   end if
26: end for

```

Data Sets	Instances	Classes	Attributes
Synthetic Data Set	1 000	2	10
Adult	48 000	2	14
Ozone	2 534	2	72
Diabetes	768	2	8
Wine Quality	6 497	10	11
Musk	475	2	166

Table 2 Characteristics of data sets used in experiments

5 Experimental Analysis

The experimental section for EVE consists of three parts. First we evaluate EVE’s ability to find deviating subspaces using synthetic data. The second exercise uses three real world data sets and evaluates how intuitive the interpretation is of the discovered subspaces. Lastly, we perform some tests on the scalability of EVE. The characteristics of the data sets used throughout the experimental analysis are described in Table 2.

Algorithm 2 EVE algorithm for finding subspaces of high dissimilarity between C and C'

Require: clusterings C and C' , full feature space A , threshold $\delta_2 < 1$

Ensure: all subspaces $A' \subset A$ in which the dissimilarity between C and C' satisfy the constraint δ_2 are returned

```

1: for  $i = 0$  to  $R$  do
2:   Calculate  $LB$  for all base subspaces
3:   Calculate  $ADCO$  for all base subspaces
4:    $SortedBase = \text{Sort values of } LB \text{ for the base subspaces}$ 
5: end for
6: for  $i = 1$  to  $R$  do
7:    $subspace = \{a_i\}$  where  $a_i \in SortedBase$  ( $a_i$  is the first element in the sorted list (has the smallest  $LB$  value))
8:   if  $LB(subspace, C, C') \leq \delta_2 \times ADCO(A, C, C')$  then
9:     if  $ADCO(subspace, C, C') \leq \delta_2 \times ADCO(A, C, C')$  then
10:      output  $subspace$ 
11:     end if
12:     Greedily select next attribute  $a_j$ 
13:     repeat
14:        $subspace = subspace \cup \{a_j\}$ 
15:       if  $LB(subspace, C, C') \leq \delta_2 \times ADCO(A, C, C')$  then
16:         if  $ADCO(subspace, C, C') \leq \delta_2 \times ADCO(A, C, C')$  then
17:           output  $subspace$ 
18:         end if
19:       else
20:          $subspace = subspace \setminus \{a_j\}$ 
21:         Backtrack to next node in enumeration tree
22:       end if
23:     until no attributes remain to be added
24:   end if
25: end for
26: end for

```

Subspaces compared by $ADCO$	$ADCO$ value
$ADCO(A', C, C')$	0.988
$ADCO(A'', C, C')$	0.504
$ADCO(A' \cup A'', C, C')$	0.549

Table 3 Synthetic Data Set: $ADCO$ similarity values between C and C' for A' , A'' and $A' \cup A''$.

5.1 Experiment A : Subspace Validation

We generated a synthetic data set having 1000 instances and formed two clusterings, $C = \{c_1, c_2\}$ and $C' = \{c'_1, c'_2\}$. Each cluster contained 500 instances and the objects were distributed to clusters differently in each clustering. A feature set $A' = \{a'_1, a'_2, \dots, a'_5\}$ was then generated, in which all values of features were randomly generated within a very close range. This enforced both C and C' to have similar distributions over A' , resulting in high similarity between them. An additional feature set $A'' = \{a''_1, a''_2, \dots, a''_5\}$ was also created, for which the clusterings C and C' were highly dissimilar. This was achieved by generating feature values using a separate range for each clustering. The two feature sets were then merged to create $A' \cup A''$ and our objective was then to use EVE to recover subspaces of

Summary	Subspaces of High Similarity	Subspaces of High Dissimilarity
Threshold value	$\delta_1 = 1.5$	$\delta_2 = 0.05$
Number of subspaces output	56	761
Number of subspaces examined	2036	1336
Total search space	1.1×10^{11}	1.1×10^{11}
Average similarity value in deviating subspace ($ADCO(A', C, C')$)	0.2	0.08
Similarity value in full feature space ($ADCO(A, C, C')$)	0.128	0.128
Time taken (seconds)	0.151	0.148

Table 4 A summary table for the Wine Quality Dataset experiment.

$A' \cup A''$, which corresponded to subspaces of A' and A'' . Table 3 shows the relative $ADCO$ values of A' , A'' and $A' \cup A''$.

EVE was able to discover the expected sets of subspaces (2^5 subspaces of A' and 2^5 subspaces of A'') belonging to A' (highly similar ones) and A'' (highly dissimilar ones) in the two discovery tasks. This exercise supports the hypothesis that when using EVE, a detailed subspace comparison between C and C' can be made. Given a single similarity value for $A' \cup A''$ (0.549), users might never be aware that there are subsets of features which give very high similarity (i.e. 0.988). Moreover, the results of the experiment indicates it may be promising to apply EVE in settings where there exist attributes that distort the clusterings in the full feature space and hence bias the user’s idea of the degree of similarity. In this case, observing subspaces may potentially provide a deeper understanding.

5.2 Experiment B : Subspace Interpretation

We took three real world data sets; ‘Wine Quality’, ‘Adult’ and ‘Ozone’ (all available from UCI Repository (Frank and Asuncion 2010)) whose characteristics are described in Table 2. We produced two clusterings of each data set and used EVE to find deviating subspaces between the clusterings. The purpose was to determine whether EVE could return intuitive subspaces when used in each of these situations.

5.2.1 Wine Quality Dataset

The Wine Quality dataset characterizes 6497 variants of a Portuguese wine by a set of measurable chemical attributes, such as pH level, alcohol measure and acidity. The dataset consists of 1599 red wine and 4898 white wine variants. The quality of each wine is also recorded by wine experts with a value between 0 and 10, where 0 indicates poor quality and 10 means excellent.

With this dataset, we are interested in identifying the subset of core attributes, which determine good or bad quality in the red wine category compared to the white wine category. Since it is difficult to achieve such a comparison easily with normal clustering comparison measures that always use the full feature space, we

Subspaces of high similarity	$ADCO(A', C, C')$ value
$A' = \{3, 8 \text{ (residual sugar, pH)}\}$	0.24
$A' = \{0, 8, 10 \text{ (fixed acidity, pH, alcohol)}\}$	0.19
$A' = \{2, 3, 8 \text{ (citric acid, residual sugar, pH)}\}$	0.18

Table 5 Subspaces where two clusterings from Wine Quality dataset are compared and found to be highly similar ($ADCO(A, C, C') = 0.127$)

Subspaces of high dissimilarity	$ADCO(A', C, C')$ value
$A' = \{4, 6 \text{ (chlorides, total sulfur dioxide)}\}$	0.068
$A' = \{6, 9 \text{ (total sulfur dioxide, sulphates)}\}$	0.053

Table 6 Subspaces where two clusterings of Wine Quality dataset are compared and found to be highly dissimilar ($ADCO(A, C, C') = 0.128$).

want to apply EVE to discover feature subspaces, in which the red wine clustering is in fact, different/similar to the white wine clustering.

We divided the dataset into white wine and red wine groups. In each group, we created a clustering of two clusters. One cluster was labeled as ‘good’ wine cluster and contained wines with scores between 7 and 10 and the other cluster was labeled as ‘bad wine cluster’ and contained wines with scores between 0 and 3. For instances with the quality index between 4 and 6, we discarded them for this experiment. We then compared the two groups (clusterings) using EVE.

Looking at the result summary in Table 4, we can see that the $ADCO$ value in the full feature space is quite low, which suggests that the spatial similarity for the underlying attributes that determine the quality between the red wine clustering and the white wine clustering is quite different. In Tables 5 and 6, we list the subspaces, where the two clusterings were highly similar and dissimilar.

When comparing the $ADCO$ value in the full feature space against the $ADCO$ values in the subspaces in Table 5, we can identify which subset of attributes has similar contribution towards the wine quality metrics and which subset of attributes has contrasting input.

For example, we can see that features such as pH, citric acid and residual sugar contribute in a similar manner for when measuring the quality of both red and white wines. On the other hand, levels of total sulfur dioxide, chlorides and sulphates tend to be differently considered in red and wine clusterings.

It is the case that the similar and dissimilar subspaces found above between the two clusterings are in fact directly correlated to the differing metrics used to measure the quality of both red and white wine. For example, wine experts tend to identify with the alcohol and fixed acidity levels of both red and white wine, when determining the quality. Moreover, since residual sugar level is inversely correlated to the acidity level, it is also considered as a determinant for wine quality measure. This supports the presence of these features in the subspaces of high similarity.

However, sulfur dioxide is a preservative added during the wine-making process, as an anti-bacterial that prevents wine from turning into vinegar. Often, one would expect to find higher amounts of sulfur dioxide in white wine, than red. This is because the tannin in red wine naturally works as a preservative, and therefore,

Summary	Subspaces of High Similarity	Subspaces of High Dissimilarity
Threshold value	$\delta_1 = 1.1$	$\delta_2 = 0.8$
Number of subspaces output	3 876	1 301
Number of subspaces examined	4 057	1 571
Total search space	8 192	8 192
Average similarity value in deviating subspace ($ADCO(A', C, C')$)	0.75	0.5
Similarity value in full feature space ($ADCO(A, C, C')$)	0.66	0.66
time taken (seconds)	0.33	0.17

Table 7 A summary table for ‘Adult’ data set experiment. Note that we show the summary in regards to comparing the countries between England and Vietnam.

there is a much less amount of sulfur dioxide added to the red wine. For higher quality white wine, one may require a larger amount of sulfur dioxide in order to preserve it longer. This supports the inclusion of sulfur dioxide in one of the subspaces with a low $ADCO$ measure between the red and white clusterings. In addition, sulphate, which is the by-product of sulfur oxide’s preservation process, would naturally occur more in the white wine than red, and it was also a member of the highly dissimilarity subspaces.

5.2.2 Adult Data Set

This data set contains information about 48 000 adults (e.g. age, education, marital status) and is used to classify whether or not an individual earns more than \$50 000 dollars (USD). For our purposes, we generated a number of clusterings, with each clustering corresponding to a particular nationality. Each clustering contained two clusters, one for those who earn more than \$50 000 and one for those who do not. Our objective was to observe the similarities and dissimilarities between clusterings. This corresponds to the differences across countries for those who earn high and low income. The data set contained 14 attributes and we removed the ‘country’ attribute before input to EVE. Table 7 shows a summary of the experiment. In Table 8 we show the comparison results between pairs of countries (clusterings).

Table 8 appears to illustrate some interesting stories between pairs of countries, in terms of how individual adults earn income and what characteristics influence their earnings. For example, when comparing the England clustering and the Vietnam clustering, we see a collection of attributes such as education level achieved (e.g. high school, tertiary), number of education (i.e. number of degrees, certificates, diplomas), the working class (e.g. self-employed, governmental) and their age, determined whether the person earned more than \$50,000 per year for both of the countries. However, collections of attributes like people’s race, relationship (e.g. wife, husband, own-child) and occupation (e.g. sales, technology, farming) were differentiating subspaces between these two countries. In the case of the comparison between the Cambodia clustering and the France clustering, we can see that these clusterings appear more dissimilar in the subspace of race, age, occupation and number of education. This suggests that this in this sub-

Countries compared	A Subspace of High Similarity	A Subspace of High Dissimilarity	$ADCO$ in Full Feature Space
England, Vietnam	education, number of education, working class, age ($ADCO = 0.81$)	race, relationship, occupation ($ADCO = 0.21$)	0.64
Japan, Jamaica	capital-gain, occupation marital status, relationship ($ADCO = 0.70$)	race, age, number of education ($ADCO = 0.29$)	0.57
Cambodia, France	capital-gain, working class, education, sex ($ADCO = 0.76$)	race, age, occupation, number of education ($ADCO = 0.3$)	0.61

Table 8 Comparing two countries in terms of how characteristics of individual adults determine their income.

Summary	Test 1 January vs. July	Test 2 1998 vs. 2004
Threshold value	$\delta_2 = 0.05$	$\delta_2 = 0.99$
Number of subspaces output	297 397	34 072
Number of subspaces examined	1.59×10^7	1 611 460
Total search space	2^{72}	2^{72}
Average similarity value in deviating subspace ($ADCO_{A'}$)	0.015	0.982
Similarity value in full feature space ($ADCO_A$)	0.330	0.983
time taken (seconds)	67	6.8

Table 9 A summary table for ‘Ozone’ data set experiment.

space, the Cambodia clustering and the French clustering have higher deviation. I.e. for this subspace, the way high-low earning capacity in France is determined appears rather different from the way high-low earning capacity in Cambodia is determined.

The results for this experiment highlight the advantage of using EVE to analyze and explore clusterings from new perspectives. Indeed, we could form clusterings for this data set in different ways. For example, generate clusterings based on different occupations and then form clusters in each based on earning level characteristics. Furthermore, the ability of EVE to compare clusterings containing non-overlapping objects (modelled by a mixture of discrete and categorical attributes) is a valuable factor in making these experiments possible.

5.2.3 Ozone Level Data Set

This data set consists of various readings relevant to monitoring the ozone level of the earth. It contains 2 534 daily observations in a 7 year span with 72 features and a class with the two values {‘ozone day’, ‘normal day’}, where the former signifies a high level of ozone (O_3) in the atmosphere and the latter indicates a normal level. We performed two tests to find subspaces of high dissimilarity.

Dissimilar Subspaces	Similarity Value ($ADCO(A', C, C')$)
$A' = \{14, 24\}$	0.977
$A' = \{13, 14\}$	0.979
$A' = \{9, 12, 24, 53\}$	0.979
$A' = \{7, 9, 10, 14, 24, 49, 53, 61\}$	0.980
$A' = \{12, 14, 15, 24, 32, 61\}$	0.980

Table 10 Comparing clusterings of ozone readings between year 1998 and 2004. Note that the similarity of two clusterings in the full feature space was 0.983.

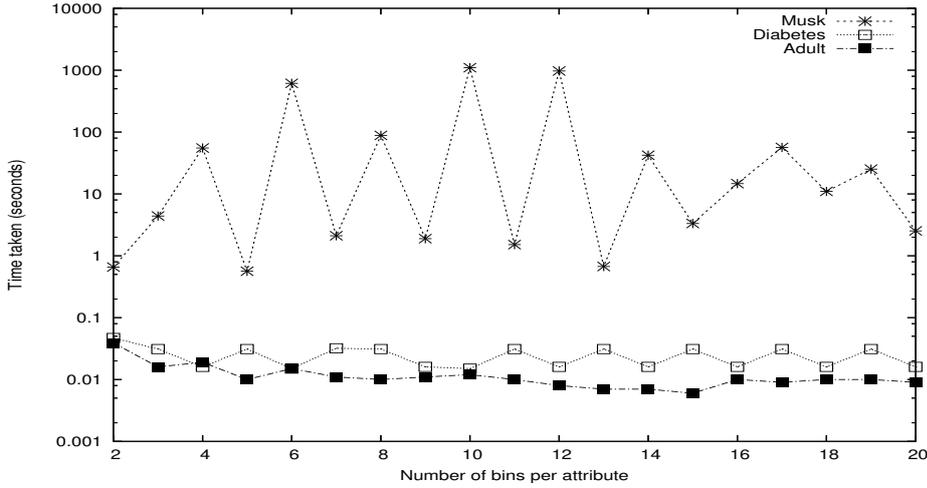
In the first test, we grouped the data into different months, combining records from the same month across all 7 years. We then formed a clustering of the data for January (with clusters ‘ozone day’ and ‘normal day’) and compared it against a clustering for July (also with clusters ‘ozone day’ and ‘normal day’). We mined dissimilar subspaces. The subspaces returned by EVE consisted of feature sets related to ‘base temperature in F° ’ from different locations, which matched our intuition, since these two months have quite different temperatures. For example, the largest subspace returned contains features $\{33, 34, 35, 36, 39, 41, 43, 44, 47, 48, 49, 50, 51\}$ (feature index numbers) where each feature is a separate reading of temperature. The similarity between the two months in the full feature space A was 0.33 while the average similarity of these dissimilar subspaces was 0.015, signifying high dissimilarity.

In the second test, we divided the data set into years and formed a clustering for each of the 7 years (each with clusters corresponding to ‘ozone day’ and ‘normal day’). The aim was to identify subspaces in which the clusterings have significantly changed. In Table 10, we show some of the subspaces that caused the biggest changes between the years 1998 and 2004. The features in this Table turn out to have names such as ‘wind speed near the sun rise’, ‘base temperature in F° ’ and ‘upwind ozone background level’. This corresponds with intuition about global warming, since this is correlated to rising temperatures and changes in ozone. Furthermore, change in wind speed is also said to be correlated with global warming (Freitas 2002). Whilst the difference between the $ADCO$ value in the subspaces and the $ADCO$ value in the full feature space is small, these subspaces might still be used to suggest a hypothesis for further exploration.

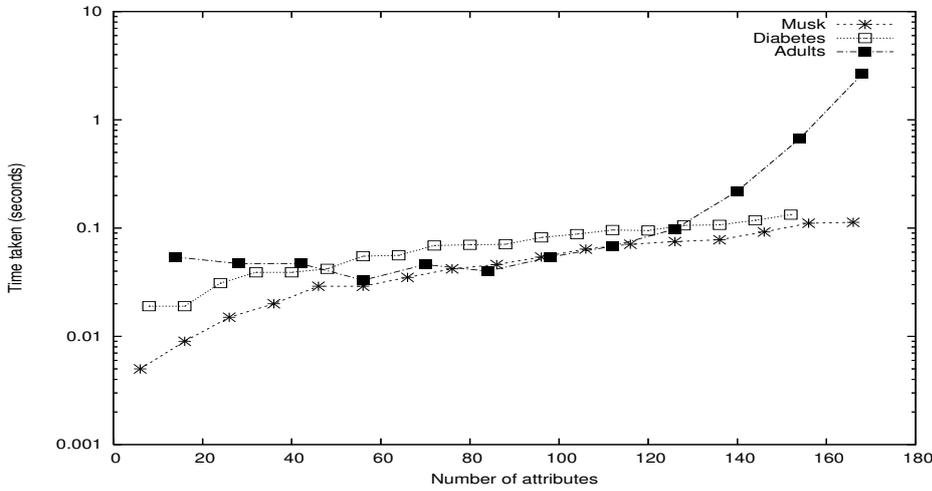
5.3 Experiment C : Pruning Effectiveness and Scalability

Information about the effectiveness of the pruning strategy used in EVE is in Table 4, Table 7 and Table 9. For example, comparing the ‘number of subspaces examined’ against the ‘total search space’ for ‘Wine Quality’ and ‘Ozone’ datasets, we see there is a dramatic difference between the two values (a factor of more than 50 million for Wine Quality and more than 100 billion for ozone).

This demonstrates that the pruning methods of EVE have an extremely significant effect on reducing the overall running time, by analyzing only a subset of exponential search space. Recall that a naive algorithm without pruning would need to explore the entire search space in order to guarantee completeness of the output answer set and for the data sets we have tested, this would be infeasible.



(a) Performance of EVE as number of bins per attribute for 'Diabetes', 'Adult' and 'Musk' varies.



(b) Performance of EVE number of attributes for 'Diabetes', 'Adult' and 'Musk' varies.

Fig. 2 Figures for testing scalability of EVE.

Since EVE employs *ADCO* as its underlying similarity measure, its overall performance is influenced by the number of attributes, bins, instances and clusters. So, using data sets 'Diabetes' and 'Musk' and 'Adult', we tested the scalability of EVE by changing these variables and the results are displayed in Figure 2(a), Figure 2(b) and Figure 3(a) respectively (characteristics of the datasets are described in Table 2).

The effect of increasing the number of bins per attribute (shown in Figure 2(a)) does not necessarily degrade the speed of EVE. Although this may hold true when we only consider the full feature space, the output of EVE is dependent upon how each $ADCO(A', C, C')$ values of subspaces is compared against $ADCO(A, C, C')$.

Therefore, based on how objects are distributed in each subspace with the given Q parameter value, the performance of EVE may differ. We note that the curve for the 'Musk' dataset in 2(a) appears to exhibit an oscillating behaviour as the number of bins increases. Further investigation revealed that this appears to be due to the choice of $\delta_2 = 0.74$ that was used in the experiment. This threshold appears to represent a tipping point in behaviour, below $\delta_2 = 0.74$ there are few subspaces output and the running time is stable around 0.2 seconds, even as the number of bins increase. Above $\delta_2 = 0.74$, the running time increases to around 1000 seconds, but is again stable as the number of bins increase.

It is also true that the effect of number of bins can be further reduced by employing not equi-width binning method, but instead implementing a more sophisticated techniques as described by Kontkanen and Myllymäki (2007). For example, the NML optimal histogram estimation adopts the minimum description length principle to compress irregular data, while emphasizing data points that occur more frequently. Furthermore, a number of density and/or entropy based feature discretization methods have also proven to be effective. For the datasets we have used in this experiment, however, the number of bins did not affect the performance or the *ADCO* values greatly and the extra step required to optimally discretize the attributes has been left as our future work.

On the other hand, growing the size of dimensionality obviously made EVE slower as shown in Figure 2(b). This is because the number of subspaces output can increase exponentially as the number of attributes increases³. However, the time required for the 'Musk' and 'Diabetes' was quite stable which affirmed the effectiveness of our pruning strategies. For the 'Adults' dataset though, there is a steep increase in running time at around 130 attributes.

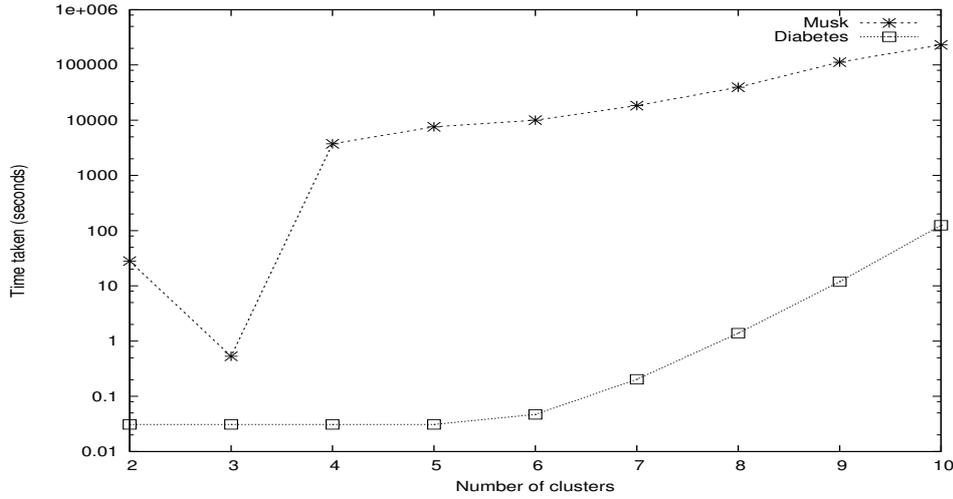
Finally, the results of EVE's performance is shown in Figure 3(a) when the number of clusters per clustering is changed. This demonstrates that running time becomes slower as the number of clusters increases. EVE is affected in such a way because the underlying *ADCO* measure requires permuting the order of clusters (using the Hungarian algorithm) in one of the clusterings, before calculating the overall dissimilarity. The implications here are that EVE is likely to be more practical for clusterings with less than 10 clusters and it may not be feasible to use for clusterings with a very large number of (e.g. 30 or more) clusters.

Additionally EVE is also affected by the δ_1 and δ_2 values. In Figure 3(b) we plot how EVE's performance is changed when we changed δ_1 value. This shows the runtime performance of EVE on the Ozone data set, as the value of $\delta_1 \times 100\%$ varies, We observe that EVE is very efficient for higher values of the threshold, but runtime increases (and pruning effectiveness decreases) as the threshold becomes lower. Intuitively, this is because the constraint is becoming less selective and the output size is becoming much larger, due to this lack of selectivity.

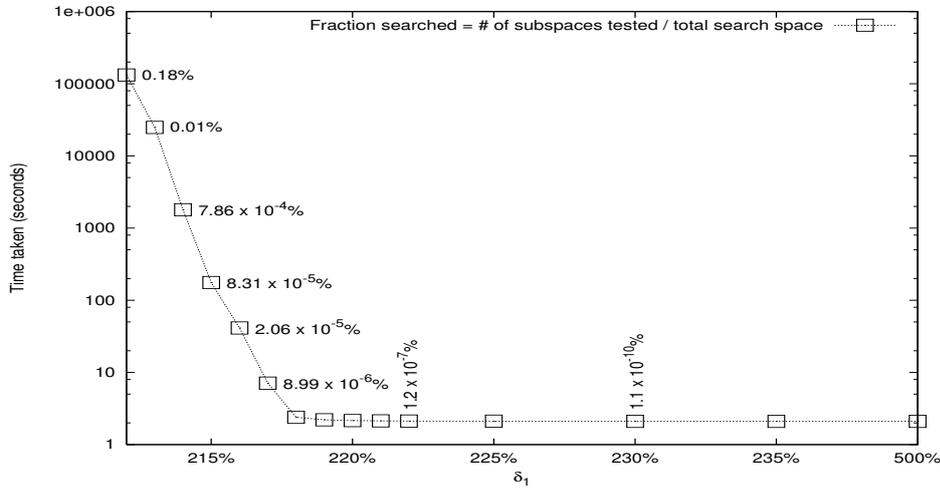
6 Discussion and Limitations

Whilst we believe that our EVE algorithm has excellent potential for exploratory clustering comparison, there remain a number of challenging issues to investigate and analyse further for the future.

³ For this experiment, we duplicated the features of 'Diabetes' and 'Adult' in order to increase the number of features up to 166



(a) Performance of EVE as number of clusters for 'Diabetes' and 'Musk' varies.

(b) Performance of EVE when δ_1 is varied for the Ozone data set.**Fig. 3** Figures for testing scalability of EVE when various parameters are changed.

Firstly, the EVE technique depends on the interpretability of the *ADCO* measure and also the interpretability of changes in the *ADCO* measure. Discussion of *ADCO*'s mathematical properties is provided by Bae et al. (2010). However, precise calibration of *ADCO* values remains an open problem. A possible direction here is to use a statistical test for assessing when the value of *ADCO* in a subspace is statistically significantly different compared to the full feature space.

Secondly, the output of *EVE* is dependent on the underlying choice of binning for the *ADCO* measure and our default in this paper has been to use 10 bins of equal density. The fact that the (absolute) similarity between two clusterings is dependent on the choice of binning can be viewed as both an advantage and a

disadvantage. The advantage is that of flexibility and the capability to incorporate domain knowledge and guidance from the user into the binning process. The disadvantage is that users who are not expert may need assistance in determining how to carry out the choice of binning. Previous work by Bae et al. (2010) has compared the effect of equi-density binning, versus equi-width, versus MDL discretization binning (Fayyad and Irani 1993) for *ADCO*. An important advantage of MDL discretization is that different attributes may have different numbers of bins and the number can be automatically determined. A general conclusion made by Bae et al. (2010) is that there is reasonable consistency in *ADCO* across different discretizations, but there do exist differences in absolute values. We highlight this as an important issue that is interesting and important to investigate further. A possible further direction would be to also investigate the use of optimal histogram density estimation (Kontkanen and Myllymäki 2007).

7 Summary

In this paper, we have introduced the problem of mining deviating subspaces in order to enrich spatially aware similarity assessment between clusterings. We also proposed an efficient algorithm for enumerating deviating subspaces, leveraging an existing similarity measure known as *ADCO*.

We believe this is an exciting new direction for clustering comparison, since it can reveal hidden relationships between the two clusterings and enrich the assessment of similarity between them.

For future work, it may be interesting to investigate whether deviating subspaces can be efficiently enumerated using alternative spatially aware clustering similarity measures. Furthermore, it may be interesting to consider the imposition of other constraints for the output set of deviating subspaces, such as number of features, or use non-redundant representations such as maximality. Efficient incorporation of these into the mining process is another interesting open question.

Acknowledgements: This research was partially supported under the Australian Research Council's Future Fellowship funding scheme (project number FT110100112).

References

- Aggarwal, C.C., Procopiuc, C.M., Wolf, J.L., Yu, P.S., Park, J.S.: Fast algorithms for projected clustering. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 61–72 (1999)
- Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the International Conference on Management of Data, pp. 94–105 (1998)
- Bae, E., Bailey, J., Dong, G.: A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings. *Data Mining and Knowledge Discovery* **21**(3), 427–471 (2010)
- Coen, M.H., Ansari, M.H., Fillmore, N.: Comparing clusterings in space. In: Proceedings of the 27th International Conference on Machine Learning (ICML), pp. 231–238 (2010)
- Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1022–1029 (1993)

- Frank, A., Asuncion, A.: UCI machine learning repository (2010). URL <http://archive.ics.uci.edu/ml>
- Freitas, C.D.: Perceived change in risk of natural disasters caused by global warming. *Australian Journal of Emergency Management* **17**(3), 34–38 (2002)
- Günemann, S., Färber, I., Müller, E., Assent, I., Seidl, T.: External evaluation measures for subspace clustering. In: Proceedings of the 20th ACM Conference on Information and Knowledge Management (CIKM), pp. 1363–1372 (2011)
- Hamers, L., Hemeryck, Y., Herweyers, G., Janssen, M., Keters, H., Rousseau, R., Vanhoutte, A.: Similarity measures in scientometric research: the Jaccard index versus salton’s cosine formula. *Information Processing and Management* **25**(3), 315–318 (1989)
- Hinneburg, A., Keim, D.: An efficient approach to clustering in large multimedia databases with noise. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 58–65 (1998)
- Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* **2**(1), 193–218 (1985)
- Kontkanen, P., Myllymäki, P.: MDL histogram density estimation. *Journal of Machine Learning Research - Proceedings Track* **2**, 219–226 (2007)
- Kuhn, H.: The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly* **2**, 83–97 (1955)
- Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 16–22 (1999)
- Meila, M.: Comparing clusterings: an axiomatic view. In: Proceedings of the International Conference on Machine Learning, pp. 577–584 (2005)
- Meila, M.: Comparing clusterings - an information based distance. *Journal of Multivariate Analysis* **98**(5), 873–895 (2007)
- Müller, E., Günemann, S., Assent, I., Seidl, T.: Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment* **2**(1), 1270–1281 (2009)
- Nagesh, H., Goil, S., Choudhary, A.: MAFIA: Efficient and scalable subspace clustering for very large data sets. Technical Report 9906-010, Northwestern University (1999). URL citeseer.comp.nus.edu.sg/328921.html
- Pei, J., Han, J., Lakshmanan, L.: Pushing convertible constraints in frequent itemset mining. *Data Mining and Knowledge Discovery*. **8**(3), 227–252 (2004)
- Pei, J., Han, J., Wang, W.: Mining sequential patterns with constraints in large databases. In: Proceedings of the International Conference on Information and Knowledge Management, pp. 18–25 (2002)
- Raman, P., Phillips, J.M., Venkatasubramanian, S.: Spatially-aware comparison and consensus for clusterings. In: Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, pp. 307–318 (2011)
- Rand, W.: Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association* **66**(336), 846–850 (1971)
- Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* **3**, 583–617 (2003)
- Tatti, N., Vreeken, J.: Comparing apples and oranges: measuring differences between exploratory data mining results. *Data Min. Knowl. Discov.* **25**(2), 173–207 (2012)
- Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research* **11**(Oct), 2837–2854 (2010)
- Wallace, D.L.: Comment. *Journal of the American Statistical Association* **78**(383), 569–576 (1983)
- Zhou, D., Li, J., Zha, H.: A new Mallows distance based metric for comparing clusterings. In: Proceedings of the International Conference on Machine Learning, pp. 1028–1035 (2005)