Anne Patrikainen

Methods for Comparing Subspace Clusterings

Licentiate's thesis submitted in partial fulfillment of the requirements for the degree of Licentiate of Science in Technology

Seattle, March 4, 2005

Supervisor: Professor Heikki Mannila

Teknillinen korkeakoulu
Tietotekniikan osasto

Lisensiaattityön tiivistelmä

Tekijä:	Anne Patrikainen
Osasto:	Tietotekniikan osasto
Työn nimi:	Menetelmiä aliavaruusklusterointien vertailuun
Title in English:	Methods for Comparing Subspace Clusterings
Professuurin koodi ja nimi:	T-61 Informaatiotekniikka
Työn valvoja:	Prof. Heikki Mannila

Tiivistelmä:

Aliavaruusklusterointimenetelmät etsivät samankaltaisten datapisteiden rykelmiä data-avaruuden eri aliavaruuksista. Nämä menetelmät yhdistelevät ja yleistävät klusterointia ja piirrevalintaa. Aliavaruusklusterointimenetelmien suosio kasvaa jatkuvasti ja uusia algoritmeja julkaistaan nopeaan tahtiin. Menetelmiä on menestyksekkäästi sovellettu muun muassa geeniekspressiodatan klusterointiin.

Monissa tärkeissä tehtävissä on olennaista pystyä mittaamaan klusterointien välisiä etäisyyksiä. Tällaisia tehtäviä ovat klusterivalidointi eli klusteroinnin laadun mittaaminen, metaklusterointi eli klusterointien joukon rakenteen tutkiminen sekä konsensusklusterointi eli useiden klusterointien yhdistäminen yhdeksi klusteroinniksi. Aliavaruusklusterointien vertailuun ei kuitenkaan ole olemassa menetelmiä eikä tavallisten klusterointien etäisyysmittoja voida tässä yhteydessä käyttää.

Tässä työssä käsittelemme aliavaruusklusterointien vertailemista. Esitämme aliavaruusklusteroinneille useita etäisyysmittoja, jotka johdetaan muun muassa yleistämällä tunnettuja tavallisten klusterointien etäisyysmittoja. Kuvaamme joukon aliavaruusklusterointien etäisyysmitoille tärkeitä teoreettisia ominaisuuksia ja tutkimme etäisyysmittojamme kattavasti näiden ominaisuuksien valossa. Todennamme etäisyysmittojemme hyödyllisyyden käytännön kokeissa, joissa vertailemme FastDOC-, HARP-, PROCLUS-, ORCLUS- ja SSPC-algoritmien tuottamia aliavaruusklusterointeja.

Sivumäärä: 83 Avainsanat: aliavaruusklusterointi, projektioklusterointi, etäisyysmitta, klusterivalidointi, metaklusterointi, konsensusklusterointi

Helsinki University of Technology
Department of Computer Science and Engineering

Abstract of Licentiate's thesis

Author:	Anne Patrikainen
Department:	Department of Computer Science and Engineering
Title:	Methods for Comparing Subspace Clusterings
Title in Finnish:	Menetelmiä aliavaruusklusterointien vertailuun
Chair:	T-61 Computer and Information Science
Supervisor:	Prof. Heikki Mannila

Abstract:

Subspace clustering methods aim to find groups of similar data points in various subspaces of the original data space. They combine and generalize clustering and feature extraction. Subspace clustering methods are becoming more and more popular, and new algorithms are being published at an increasing rate. These algorithms have been successfully applied for instance to gene expression data.

Having distance measures for clusterings is essential in many important tasks. These measures are needed in cluster validation, i.e., in measuring the quality of a clustering; in meta-clustering, i.e., in examining the structure of a set of clusterings; and in consensus clustering, i.e., in combining several clusterings into a single clustering. There are no existing methods for comparing subspace clusterings, and the methods for comparing ordinary clusterings are not applicable to this task.

We present the first framework for comparing subspace clusterings. We propose several distance measures for subspace clusterings, including generalizations of well-known distance measures for ordinary clusterings. We describe a set of important properties for any measure for comparing subspace clusterings. We give a systematic comparison of our proposed measures in terms of these properties. We validate the usefulness of our subspace clustering distance measures by comparing clusterings produced by the algorithms FastDOC, HARP, PROCLUS, ORCLUS, and SSPC.

Number of pages: 83 Keywords: subspace clustering, projected clustering, distance measure, cluster validation, meta-clustering, consensus clustering

Preface

This thesis has been written after two years of graduate studies in the Laboratory of Computer and Information Science at the Helsinki University of Technology. It has been a good place to work at — I really feel at home there. I am very grateful to my advisor Prof. Heikki Mannila: it has been an honor to work with a top-class scientist and see how good research is done. I have enjoyed the meetings of our Pattern Discovery research group; Antti, Antti, Ella, Heli, Jaakko, Janne, Jaripekka, Johan, Jouni, Kalle, Nikolaj, and Salla have been great company. I have had the privilege of sharing a room with the best possible officemate, Johan. Antti H. and Jouni have provided much appreciated technical support on countless occasions. Finally, special thanks to Prof. Erkki Oja whose empathetic leadership skills I truly admire.

The bulk of the research presented in this thesis was done during my 9-month visit at the University of Washington in Seattle in 2004. Once again thanks to Prof. Heikki Mannila for providing me the funding for this great opportunity to learn numerous things about research and about life. At UW, I was fortunate to be advised by Prof. Marina Meila, whose encouragement has been valuable. The days in Seattle were made happier by Pradeep, Krzysztof, Matt, Parag, and Deepak. And lucky I was also with my always cheerful UW officemates Anna, Benson, Shobhit, and Sushant.

I would like to acknowledge Kevin Yip from the University of Hong Kong; he has kindly provided me the subspace clusterings I have used in the experimental results section of this thesis. Lots of thanks to Kevin also for the numerous inspiring discussions we have had about internal cluster validation.

Panayiotis Tsaparas has acted as the external reviewer of this thesis. I appreciate the time he has spent in carefully going through the text and giving valuable comments.

Huge heartfelt thanks go to Antti S., Lev, Nikolaj, Riitta, and Sari, who have supported me during difficult times and changed my life for the better.

Lastly, I would like to thank my mother Mirja for teaching me to work hard; my father Tapsa for teaching me that also other things than work are important in life; and my brother Antti for showing me in practice that I do not need to accept everything that my parents teach me.

Otaniemi, December 12, 2004

Anne Patrikainen

Contents

Chapter 1

Introduction

1.1 Subspace Clustering

The goal of clustering is to group a given set of data points into *clusters* that capture some notion of similarity between the data points in each cluster. Data is represented by a number of *features*, not all of which are useful for comparing individual data points. In particular, the choice of the set of features used to represent data may highlight different facets of the similarity between the data points. Subspace clustering was introduced in order to capture this idea of "similarity examined under different representations".

Conceptually, subspace clustering algorithms work on a collection of data points described using a large number of features, and simultaneously address the problem of selecting the relevant features, and the points that are similar given these features. Formally, a *subspace cluster* can be defined as a pair (subset of data points, subspace). The data points of the cluster are similar in the associated subspace. A *subspace clustering* is a collection of subspace clusters.[1] The first[2] subspace clustering algorithm CLIQUE [4] was published in 1998 and was soon followed by many related methods [1, 2, 3, 9, 13, 14, 15, 16, 19, 20, 26, 33, 39, 41, 44, 49, 51, 55, 63, 66, 67]. The algorithms have been applied for instance to clustering gene expression data: it is often the case that a group of genes behaves similarly only in a subset of experiments (i.e. in a subspace) [15, 16, 20, 26, 55, 63, 66]. Reviews of some of the existing subspace clustering algorithms can be found in [47, 48, 68].

[1] Other names that have been used for the same or a closely related task are projected clustering [2, 3, 66, 67], projective clustering [1, 51], bi-clustering [36, 36], co-clustering [9, 16, 18], coupled two-way clustering [27], simultaneous clustering [50], direct clustering [31], block clustering [31], and clustering on subsets of attributes [26].

[2] In fact, related ideas had been introduced earlier in [31, 45], but CLIQUE was the first algorithm that became widely known in the research community.

1.2 Comparing Subspace Clusterings

Surprisingly, despite the multitude of subspace clustering algorithms, there are no existing methods for comparing their outputs. Pairs of ordinary clusterings (partitions of the set of data points) can be compared with numerous well-known criteria, for instance with the Clustering Error (CE), the Variation of Information (VI), or the Rand index; some of these measures have been in use at least since the sixties [56]. However, these criteria are not directly applicable to comparing subspace clusterings. This is unfortunate, since clustering comparison methods are necessary for several important tasks, including cluster validation, meta-clustering, and consensus clustering; we introduce these in Section 1.3.

To the best of our knowledge, nobody has proposed a method for comparing two subspace clusterings in a way that takes into account the data point groups and the subspaces simultaneously. In the existing literature, authors most commonly compare only the grouping of data points into clusters, ignoring the similarity or dissimilarity of the associated subspaces [3, 20, 51, 55, 68]; note that this approach does not work in the general case, as we will see in Chapter 3. Sometimes the subspaces are compared and the data point groups ignored [47, 49]; this is done qualitatively in the absence of suitable comparison methods. At best, the data points and the subspaces are compared separately, and the conclusions are once again only qualitative [2, 15]. All these approaches fail to compare subspace clusterings in a fair manner.

In this thesis, we present a framework for comparing subspace clusterings by generalizing well-known distance measures for ordinary clusterings. We start by introducing a set of important properties for a subspace clustering distance measure. We then propose four candidate distance measures for subspace clusterings. We characterize our candidates in terms of these theoretical properties and evaluate them experimentally.

There are various types of subspace clusters and subspace clusterings. Perhaps the most common type of subspace cluster is an *axis-aligned subspace cluster*, in which the subspace is spanned by a subset of the attributes. In this case, an equivalent representation for the cluster is a pair (subset of data points, subset of attributes). A more general type of subspace cluster is a *non-axis-aligned subspace cluster*, in which the subspaces can be arbitrarily oriented. We present distance measures for both types of subspace clusterings and also address the closely related topics of *co-clusterings* and *attribute weighted clusterings*. These categories of algorithms are described in detail in Section 2.2.

1.3 Applications of Comparing Clusterings

1.3.1 Cluster Validation

Cluster validation refers to quantitatively evaluating the quality of a clustering solution. It is always important to validate the clustering solution after running a clustering algorithm, since bad clusterings arise in many common occasions. Most clustering algorithms give some kind of a result even if the data set does not have any clustering structure. Many algorithms give bad results if the choice of the parameter values or the initialization has not been successful. This kinds of clusterings can be recognized and avoided by means of cluster validation.

Cluster validation can be divided into external and internal cluster validation. External cluster validation refers to comparing a clustering solution to a true clustering; internal cluster validation evaluates the clustering result without any knowledge of a true clustering. We briefly introduce both types of cluster validation here; a more comprehensive review can be found in Chapter 3.

External cluster validation is important in evaluating the performance of a clustering algorithm on synthetic data sets. It aims to measure the quality of clustering produced by the algorithm by comparing it to a true clustering of the data. Specifically, assume that we have a data set X for which the true subspace clustering $\mathcal{T}$ is known. We have two algorithms, A and A', which have produced subspace clusterings $\mathcal{S}$ and $\mathcal{S}'$. We would like to be able to calculate the distances $d(\mathcal{S}, \mathcal{T})$ and $d(\mathcal{S}', \mathcal{T})$ to find out which of these clusterings is closer to the true clustering. However, as argued in the previous section, there are currently no distance measures for subspace clusterings.

Internal cluster validation aims to measure the quality of a clustering in real-life settings, when there is no knowledge of the real clustering, or if there is uncertainty of whether the data set can be clustered at all [37, 56]. Internal cluster validation can be done by means of *point configuration based methods*, which assess the structure of the clustering — for instance, a clustering with compact, spherical, well-separated clusters might be judged good. Another way to do internal cluster validation is to use *stability based methods*, which measure the stability of a clustering by sampling the data. If the clusterings on different samples agree, the clustering is judged stable and therefore good. Naturally, clustering distance measures are needed for evaluating the agreement of the clusterings on different samples. However, the point configuration based methods do not need clustering distance measures, but as we will argue in Section 3.2.3, these methods are not currently applicable to subspace clusterings in the first place.

1.3.2 Consensus Clustering

If we have several clusterings for the same data set, we might be interested in combining these clusterings into a single *consensus clustering*[3] which should be as close to all original clusterings as possible [28, 54, 59, 60, 61]. This kind of situation might arise if we are not sure which parameter values (such as the number of clusters K) of a given clustering algorithm to use; we might just run the algorithm for several choices of the parameter values and try to find a consensus among the resulting clusterings. Also, combining several results by the same algorithm could alleviate the effect of random initialization (for instance choosing the initial cluster centroids).

As another possibility, we might run several clustering algorithms with different objective functions for the same data set and combine all resulting clusterings into a single clustering. This might allow us to capture a rich variety of features which would not be possible for any single clustering algorithm alone. Further, combining the results of several different algorithms would improve the robustness and stability of clustering and decrease the sensitivity to outliers and noise.

Clustering ensembles can also be used to combine clusterings on different attributes of the same data set. One potential application of this would be parallelization; another is distributed data mining. Yet another application would be clustering categorical data, where each categorical attribute could be viewed as a clustering of the data set. [28, 59]

Many of the above cases apply to subspace clusterings in addition to ordinary clusterings. Naturally, if we wish to find a consensus clustering which is close to all the original clusterings, we must be able to calculate the distances between the clusterings. Given this definition of a good consensus clustering, having a distance measure for subspace clusterings is essential for finding a consensus subspace clustering. Of course, different definitions for a consensus clustering cost function might be possible, and clustering distance measures might not always be necessary.

1.3.3 Meta-clustering

Meta-clustering refers to investigating the structure of a set of clusterings. Meta-clustering discards the idea of trying to derive a single good clustering for a data set; instead, it is acknowledged that the data can be well represented in several different, complementary ways. For instance, assume that a given data set has been clustered several times by different algorithms. A meta-clusterer might now observe that these clusterings form two tight groups of

[3] Also known as a clustering ensemble, or an aggregate clustering.

clusterings, and give the user a representative of each of these groups, instead of a single 'best' clustering. [7]

There are various ways to produce different clusterings for a data set: we could use different algorithms, a single algorithm with various parameter values and initializations, change metrics, use various dimensionality reduction schemes, or sample the data. Meta-clustering may be used to investigate whether some of these clusterings form tight groups, whether some of the clusterings are outliers, whether the effect of the parameter values is strong or weak, etc. For instance, it has been empirically shown by means of meta-clustering that only a small number of clustering algorithms is enough to represent a large number of clustering criteria [32].

Sometimes meta-clustering is used in a broader sense to refer to all kinds of methods that operate on sets of clusterings; according to this definition, meta-clustering would include cluster validation and consensus clusterings as special cases [28]. Whichever definition we choose, it is clear that meta-clustering is impossible without a distance measure for clusterings.

1.4 Contributions of the thesis

In this thesis, we address the problem of comparing subspace clusterings. Specifically,

- we introduce a set of important properties for a subspace clustering distance measure;
- we propose four novel subspace clustering distance measures (RNIA and generalizations of the Rand index, CE, and VI);
- we describe how our distance measures can be applied to axis-aligned subspace clusterings, non-axis-aligned subspace clusterings, attribute weighted clusterings, and co-clusterings;
- we investigate the theoretical properties of the proposed distance measures;
- we show experimentally that our distance measures are useful in practice.

As discussed above, subspace clustering distance measures are necessary in external cluster validation (comparing a clustering to a true clustering), stability based internal cluster validation (evaluating the stability of a clustering), meta-clustering (investigating the structure of a set of clusterings), and many cases of consensus clustering (finding a representative clustering for a set of clusterings).

In addition to comparing subspace clusterings, our distance measures can be used to compare partial clusterings (clusterings on subsets of data points),

clusterings with overlapping clusters (a data point may belong to multiple clusters), hierarchical clusterings, and all combinations of these types of clusterings.

1.5 Structure of the thesis

In Chapter 2, we define and motivate the task of subspace clustering. We also present a classification and examples of subspace clustering algorithms and related algorithms. Chapter 3 contains a detailed survey of existing methods for comparing ordinary clusterings and an overview of cluster validation methods in general. The main contributions of the thesis are presented in Chapters 4 and 5. To keep the presentation simple, we start by comparing axis-aligned subspace clusterings in Section 4. We will extend our analysis to the case of non-axis-aligned subspace clusterings, attribute weighted clusterings, and co-clusterings in Chapter 5. In Chapter 6, we apply our distance measures to comparing subspace clusterings produced by the algorithms FastDOC, HARP, PROCLUS, ORCLUS, and SSPC on synthetic data sets. Finally in Chapter 7, we present a summary of our work and discuss future research directions.

Chapter 2

Subspace Clustering

In this chapter, we define and motivate the problem of subspace clustering, introduce different types of subspace clusterings and related clusterings, and present a summary of several popular subspace clustering algorithms.

2.1 Basic Concepts of Subspace Clustering

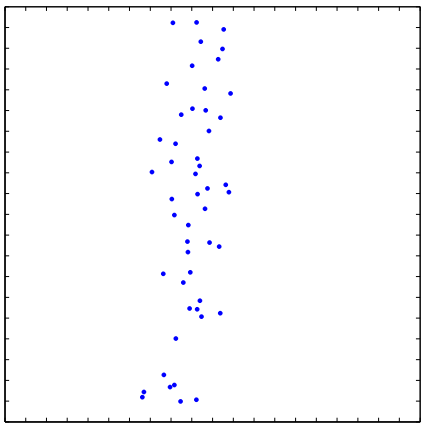
In high dimensional spaces[1], traditional clustering methods suffer from the curse of dimensionality, which is why their application is often preceded by feature selection and extraction. For instance, a practitioner might apply Principal Component Analysis (PCA) to project the data onto a low-dimensional subspace before trying to cluster the data points. However, it is sometimes unrealistic to assume that all clusters of points lie in the same subspace of the data space. Subspace clustering methods address this issue by assigning a distinct subspace to each group of data points.

Before proceeding, let us introduce our notation. The data matrix X consists of elements $x_{ij} \in \mathbb{R}$, where $i \in \{1, 2, \ldots, m\}$ and $j \in \{1, 2, \ldots, p\}$. We denote the m rows by $\{r_1, r_2, \ldots, r_m\}$, where $r_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, and the p columns by $\{c_1, c_2, \ldots, c_p\}$, where $c_i = (x_{1i}, x_{2i}, \ldots, x_{mi})^T$. We will often refer to the rows as data points and to the columns as attributes. A *cluster* $C_i \subseteq \{r_1, r_2, \ldots, r_m\}$ is a subset of the data points. A *clustering*[2] $\mathcal{C}$ is a partitioning of the set of m data points into clusters $C_1, C_2, \ldots, C_K$ of sizes $m_1, m_2, \ldots, m_K$, where $\sum_i m_i = m$.

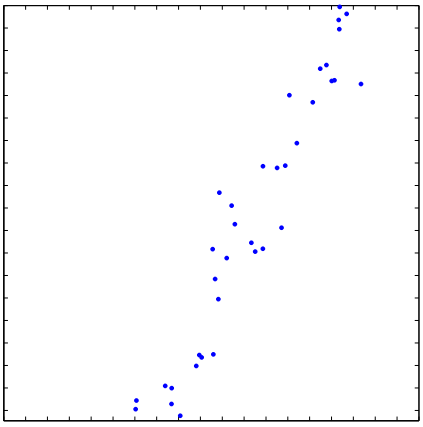
We can view a data point, or a row of the data matrix, as a vector in $\mathbb{R}^p$. A set of vectors $\mathcal{F}$ is a *subspace* of $\mathbb{R}^p$ if for any vectors v_i, $v_j \in \mathcal{F}$ and any

[1]Subspace clustering algorithms are typically run on data sets of some hundreds of dimensions.

[2]In this thesis we only discuss *hard clusterings*. In a *soft clustering*, a given data point r_i has a probability $P(r_i|C_j)$ of belonging to a given cluster C_j.



(a) Axis-aligned subspace cluster.



(b) Non-axis-aligned subspace cluster.

Figure 2.1: Two examples of a subspace cluster. Traditional clustering algorithms will not find clear clusters in these two-dimensional data sets. However, if the points are projected onto an appropriate one-dimensional subspace, a compact cluster emerges in each case. In (a), the appropriate subspace is the x-axis, but in (b), the subspace is not aligned along the axes.

$a \in \mathbb{R}$ it holds that $v_i + v_j \in \mathcal{F}$ and that $av_i \in \mathcal{F}$. A *subspace cluster* is a pair $(R, \mathcal{F})$, where $R \subseteq \{r_1, r_2, \ldots, r_m\}$ is a subset of the rows and $\mathcal{F}$ is a subspace of $\mathbb{R}^p$. A *subspace clustering* $\mathcal{S} = \{S_1, S_2, \ldots, S_K\}$ is a collection of subspace clusters.

Fig. 2.1 presents two examples of a subspace cluster. In the two-dimensional data sets, no clustering structure is visible, but if the data is projected onto a 1-dimensional subspace, a compact cluster emerges. However, in general, using feature extraction/selection before clustering is not enough to find the subspace clusters. As an example, Fig. 2.2 illustrates that PCA is of no use in solving the subspace clustering problem. The three-dimensional data set contains three subspace clusters with orthogonal 1-dimensional subspaces, but PCA is not able to reduce the dimensionality of the data.

2.2 Subspace Clusterings and Related Clusterings

We next introduce two types of subspace clusterings and two related types of clusterings. We present methods for comparing clusterings from all these categories in Chapters 4 and 5.

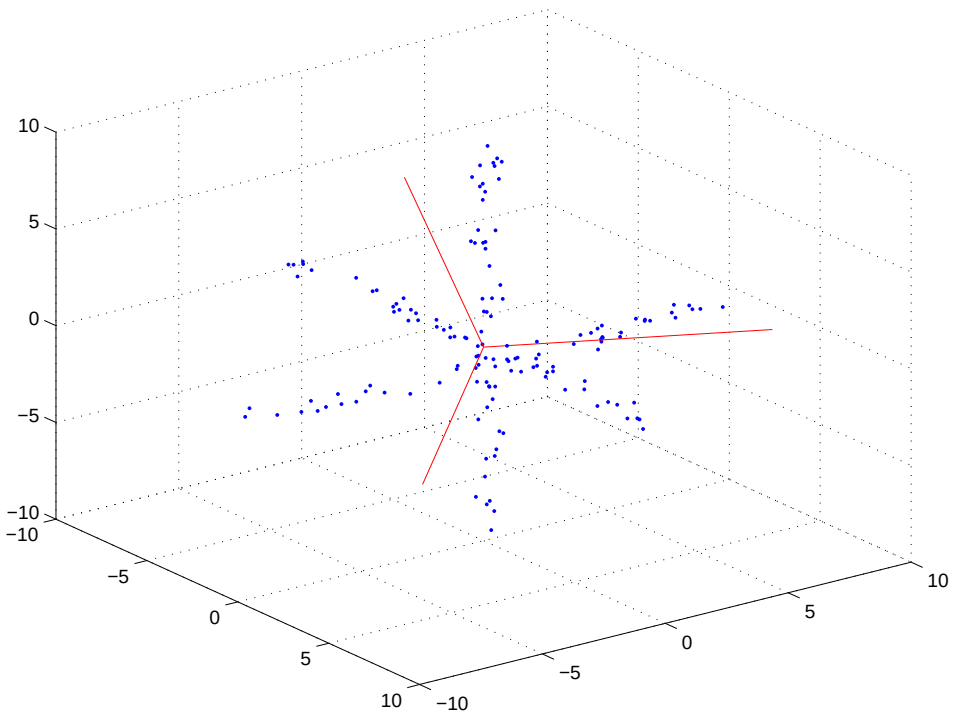


Figure 2.2: PCA cannot be used to solve some simple cases of subspace clustering. These three-dimensional data points would form clear subspace clusters on each of the xy-, xz-, and yz-planes, but the principal components (solid red lines) cannot identify the correct subspaces, nor can they be used to reduce the dimensionality of the data.

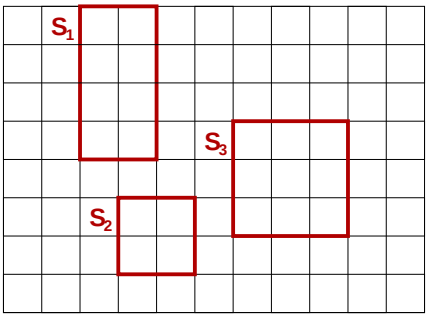


Figure 2.3: An example of an axis-aligned subspace clustering $\mathcal{S} = \{S_1, S_2, S_3\}$. The clusters are submatrices of the data matrix. In general, visualization of a subspace clustering is more difficult: It is often not possible to find a permutation of the rows and the columns such that the visual structure of the submatrices is preserved.

2.2.1 Axis-Aligned Subspace Clusterings

An *axis-aligned subspace cluster* S is a pair (R, C), where $R \subseteq \{r_1, r_2, \ldots, r_m\}$ is a subset of the rows and $C \subseteq \{c_1, c_2, \ldots, c_p\}$ is a subset of the columns. An axis-aligned subspace cluster can be conveniently viewed as a submatrix of the data matrix X (see Fig. 2.3 for an example). However, we do not consider a single element of the data matrix as a cluster.[3]

An *axis-aligned subspace clustering* $\mathcal{S}$ is a collection $\{S_1, S_2, \ldots, S_K\}$ of K subspace clusters, where $K \geq 1$. The row and column sets of the subspace clusters may overlap in an arbitrary way. We do not require that the row sets cover the set of all rows $\{r_1, r_2, \ldots, r_m\}$; we do not have this kind of requirement for the column sets either. Examples of axis-aligned subspace clustering algorithms include PROCLUS [2], CLIQUE [4], ENCLUS [14], the algorithm by Cheng&Church [15], SUBCLU [33], MAFIA [44], DOC [51], and SAMBA [55]. Axis-aligned subspace clustering is a generalization of feature selection; instead of having a single set of features (attributes) for the whole data, we have a distinct feature set for each cluster.[4]

2.2.2 Non-Axis-Aligned Subspace Clusterings

A *non-axis-aligned subspace cluster* S is a pair (R, W), where $R \subseteq \{r_1, r_2, \ldots, r_m\}$ is a subset of the rows and W is a collection of vectors $\{w_1, w_2, \ldots, w_D\}$, where

[3]The reason for this becomes clear in Section 4.2 in which we discuss backgroud independence, an important property for subspace clustering distance measures.

[4]In this context, feature refers to a combination of the attributes.

$w_i \in \mathbb{R}^p$. The vectors in W form a basis for an arbitrary subspace of the original p-dimensional data space. We use W also to denote this subspace. Naturally, an axis-aligned subspace cluster is a special case of a non-axis-aligned subspace cluster. In the case of an axis-aligned subspace cluster, W is a subset of the original basis vectors $\{e_1, e_2, \ldots, e_p\}$, where $e_1 = (1\,0\,0\,\ldots\,0)$, $e_2 = (0\,1\,0\,0\,\ldots\,0)$, etc.

A *non-axis-aligned subspace clustering* $\mathcal{S}$ is a collection $\{S_1, S_2, \ldots, S_K\}$ of K non-axis aligned subspace clusters. The algorithms ORCLUS [3], KSM [1], and 4C [13] produce these kinds of clusterings. Non-axis-aligned subspace clustering is a generalization of feature extraction; instead of defining a single set of features for the whole data, we have a distinct set of features for each cluster.

2.2.3 Attribute Weighted Clusterings

An *attribute weighted cluster* S is a pair (R, b), where $R \subseteq \{r_1, r_2, \ldots, r_m\}$ is a subset of the data points (rows) and b is a vector $[b_1, b_2, \ldots, b_p]^T$ with $b_i \geq 0$ for each $i = 1, 2, \ldots, p$, and $\sum_{i=1}^{p} b_i = 1$. The vector b defines an importance weight for each attribute (column). The importance weight is used for calculating the distances between the data points in the same cluster. For instance, for data points $r_i, r_j \in R$, we could calculate the weighted L_2 distance of the corresponding data vectors by $(\sum_{k=1}^{p} b_k(x_{ik} - x_{jk})^2)^{1/2}$.[5]

An *attribute weighted clustering* $\mathcal{S}$ is a collection $\{S_1, S_2, \ldots, S_K\}$ of K attribute weighted clusters. The algorithms LAC [20] and COSA [26] produce this type of clusterings. Attribute weighted clustering is a generalization of attribute (feature) weighting; instead of defining a single weight vector for the whole data set, we obtain a distinct weight vector for each cluster.

2.2.4 Co-Clusterings

A *co-clustering* $\mathcal{S} = (\mathcal{R}, \mathcal{C})$ consists of a partition $\mathcal{R} = \{R_1, R_2, \ldots, R_L\}$ of the rows and a partition $\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$ of the columns of the data matrix. In a co-clustering, the whole data matrix is partitioned into subspace clusters $S_{ij} = (R_i, C_j)$, where $i \in \{1, 2, \ldots, L\}$, and $j \in \{1, 2, \ldots, M\}$. A co-clustering can be visualized as a rectangular partition of the data matrix elements. Co-clustering algorithms are presented for instance in [9, 16, 18]. For an example of a co-clustering, see Fig. 2.4.

[5] In the presence of importance weights, defining the distance between data points of different clusters is not easy, and the definition depends on the algorithm.

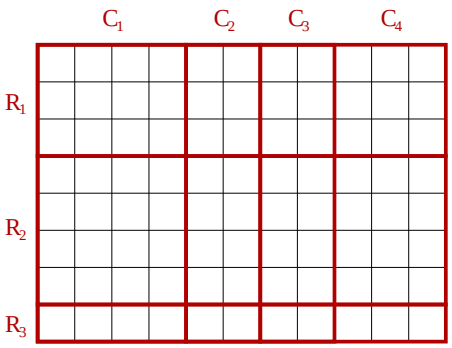


Figure 2.4: An example of a co-clustering $\mathcal{S} = (\{R_1, R_2, R_3\}, \{C_1, C_2, C_3, C_4\})$.

2.3 Examples of Subspace Clustering Algorithms

We next present examples of well-known algorithms from all four categories described above. Many of these algorithms will be compared in the experiments in Chapter 6.

2.3.1 CLIQUE

CLIQUE [4] was the first subspace clustering algorithm, published in 1998. CLIQUE is a grid-based clustering algorithm, and the clusters are found by an apriori-style bottom-up search technique. The clusters produced by CLIQUE are unions of hypercubes in axis-parallel subspaces of the original space. CLIQUE was rapidly followed by new algorithms ENCLUS and MAFIA which extended its basic idea [14, 44].

CLIQUE attempts to find subspace clusters in a high-dimensional continuous-valued data set. A given subspace of the data space is partitioned into hypercubes whose sizes are given to CLIQUE as an input parameter. The density of a hypercube is defined as the number of data points in it (there is no need to divide by the size of the cube, as the cubes are of equal size). If the density of a hypercube exceeds a user-defined threshold, the hypercube is considered a cluster, or a part of a cluster, in that subspace. The clusters in a given subspace are defined as unions of adjacent dense hypercubes.

CLIQUE starts by locating the clusters in one-dimensional subspaces, after which it proceeds to higher-dimensional subspaces. The higher-dimensional subspaces with no potential clusters are pruned away. Despite this, the number of candidate subspaces might grow too large, and some additional subspaces with small *coverage* are pruned away. The coverage is defined as the fraction of the data points in the clusters; the pruning is done with a MDL-type criterion.

CLIQUE is able to find any number of clusters of various shapes (arbitrary unions of adjacent hypercubes) and of any dimensionality. However, the choice of the input parameters (hypercube size and density threshold) has a strong effect on the results. Further, CLIQUE easily discards clustered data points as outliers, for instance on the outer borders of spherical clusters.[2]

CLIQUE produces axis-parallel subspace clusters; its output clusters are usually *non-disjoint*: a pair (data point, dimension) may belong to several clusters of a CLIQUE clustering.

2.3.2 PROCLUS

PROCLUS [2] (PROjected CLUStering) is an iterative algorithm reminiscent of K-medoids [35]. In each iteration, the subspaces associated with the clusters are updated and the data points are redistributed across the clusters. PROCLUS was one of the first subspace clustering algorithms, published right after CLIQUE. PROCLUS produces axis-aligned subspace clusters.

PROCLUS is initialized by picking a large set of candidate medoids (data points that act as cluster representatives), of which K are chosen at random. If some of these K medoids prove to be bad choices, they are replaced by new ones from the set of candidate medoids. The data is partitioned into K groups by assigning the data points to the closest medoids (at the later stages of the algorithm, these distances are calculated in the appropriate subspaces). Each medoid is then associated with a subspace. A given dimension is considered relevant for a given cluster if the variance of the cluster data points is sufficiently small along that dimension. Finally, the medoids for the data point groups are re-calculated. The medoids with only a few data points assigned to them are replaced by new ones from the set of remaining candidate medoids. These steps are repeated until the algorithm converges. After the convergence, some points are discarded as outliers and the clusters are reported to the user.

PROCLUS requires two input parameters: the number of clusters K and the average cluster dimensionality L. The sum of the subspace dimensionalities in the output clustering is then restricted to KL. This is a severe restriction, as the user has often no idea which parameter values to choose, and unfortunately, the effect of the parameter values on the results is large. On the other hand, the authors of PROCLUS claim that PROCLUS is an order of magnitude faster than CLIQUE, and that CLIQUE discards about half of the relevant points as outliers, whereas PROCLUS is able to identify most of the relevant points.

2.3.3 ORCLUS

ORCLUS [3] (arbitrarily ORiented projected CLUSter generation) was published right after PROCLUS by the same researchers, and the two algorithms are similar in many respects. The main difference is that ORCLUS is able to handle non-axis-aligned subspaces — in fact, it is the first subspace clustering algorithm capable of that.

ORCLUS is based on the popular K-means algorithm [35]. First, K centroids are chosen at random to represent the clusters. The data points are then partitioned by assigning them to the closest centroids (at the later stages of the algorithm, the distances are computed in the appropriate subspaces). Each group of data points is associated with a subspace; these subspaces are found with a variant of PCA. A covariance matrix is computed for each group of data points separately. Since subspaces with small data variance are desired, the eigenvectors corresponding to the smallest eigenvalues of the cluster data covariance matrix are utilized instead of the largest ones. Finally, the centroids are replaced by the means of the cluster data points. These steps are repeated until the convergence of the algorithm. ORCLUS also contains a merge stage, in which two clusters with similar subspaces can be merged if necessary.

ORCLUS has two input parameters: the number of clusters K and the desired dimension of the subspaces D_s. This means that the subspaces are restricted to have the same pre-defined dimension. ORCLUS is computationally heavy due to the calculation of numerous covariance matrix eigendecompositions.

2.3.4 DOC and FastDOC

DOC (Density-based Optimal projective Clustering) and its variant FastDOC are Monte Carlo algorithms that produce a single axis-aligned subspace cluster at a time [51]. A subspace cluster is defined as a dense hypercube of width 2ω, where ω is an input parameter. The quality of a cluster is evaluated by an objective function that takes into account the number of data points in the hypercube and the dimensionality of the hypercube. The more data points and relevant dimensions a cluster has, the less likely it is to have been formed by chance, and the better is the score it gets. The relative importance between the number of points and the dimensions is controlled by another input parameter β. A third parameter α is needed to specify the minimum number of data points that can form a cluster.

At first, a random data point is selected as a seed for a subspace cluster. More data points are sampled in the vicinity of these seeds in order to de-

termine the relevant subspace of the cluster. The algorithm repeatedly tries different seeds until the maximum number of iterations is reached. The cluster with the highest score is returned, and the algorithm may proceed to finding a new cluster.

DOC and FastDOC require three input parameters ω, α, and β that might be hard to specify by the user; yet they have a strong effect on the results. On the other hand, selecting the number of clusters K is not as big a problem with DOC and FastDOC as with most other clustering algorithms, since (Fast)DOC produces one cluster at a time.

2.3.5 HARP

HARP [65, 66] (a Hierarchical approach with Automatic Relevant dimension selection for Projected clustering) is an agglomerative hierarchical subspace clustering algorithm based on greedy merging of the most similar clusters. At first, each data point forms its own subspace cluster. A merge score is calculated between each pair of clusters, and the cluster pair with the best score is merged. This process is repeated until all data points are in the same cluster.

The subspace clusters produced by HARP are axis-aligned. A given dimension is considered relevant for a cluster if the cluster data variance is sufficiently low compared to the global data variance on that dimension. (Naturally, all dimensions are relevant for the initial clusters with one point each.) A pair of clusters receives a good merge score if the two clusters are similar in a number of relevant dimensions. The minimum similarity (related to the variance of the data points) and the minimum number of similar dimensions are controlled by two internal threshold variables. HARP automatically adjusts the values of these parameters during the clustering process.

HARP is completely parameter-free and deterministic, and therefore very straightforward to use. As an hierarchical clustering algorithm HARP produces a dendrogram of the subspace clusters as its output. However, the user has the option of giving the desired number of clusters as an input, in which case HARP outputs a flat clustering.

2.3.6 SSPC

SSPC [67] (Semi-Supervised Projected Clustering) is the first semi-supervised subspace clustering algorithm. The pracititioner can use his domain knowledge to improve the clustering accuracy of SSPC by supplying labeled objects and labeled dimensions as input parameters. However, SSPC can be used also in a fully unsupervised manner. The authors claim that SSPC is particularly

suitable for identifying extremely low-dimensional subspace clusters, with dimensionalities as small as 5% of the total number of dimensions. SSPC is an axis-aligned subspace clustering algorithm.

Unlike most other subspace clustering algorithms, SSPC is designed to optimize the value of an objective function, which reflects the data distribution model central to the algorithm. The main assumption is that the data corresponding to the relevant dimensions of a subspace cluster is sampled from a Gaussian distribution with a small variance, and that the data corresponding to the irrelevant dimensions is sampled from a Gaussian distribution with a large variance.

The algorithm itself is iterative. First, a set of candidate seeds (cluster representatives) is determined, and K of these are selected to initialize the cluster locations. In each iteration, the objects are assigned to the clusters that give the greatest improvement to the objective score, after which the subspaces are updated. If the objective score has improved, the clustering is saved, otherwise the old clustering is restored. If some of the clusters is bad, a new seed will be selected for it. Finally, the cluster centroids are updated.

The parameters required by SSPC are the desired number of clusters K together with a parameter that controls the subspace updating procedure. Additionally, the user is free to supply a set of labeled data points and attributes. The experimental results demonstrate that the accuracy of the algorithm is improved when prior knowledge is incorporated in the clustering process. SSPC is designed to handle noise as well.

2.3.7 LAC

LAC [20] (Locally Adaptive Clustering) is an attribute weighted clustering algorithm. Its output clustering is a partitioning of the set of data points, where each group of data points is associated with a weight vector. The weights are larger for more relevant dimensions. Given a group of data points and a dimension d, the weight is large if the variance of the data points along that dimension is small.

LAC is a K-means based iterative algorithm that resembles PROCLUS and ORCLUS. An iteration starts by assigning each data point to the closest cluster; the dimensions are weighted by the cluster-specific weight vectors. The weights are updated next, after which the data points are assigned to the closest clusters again. Finally, new centroids are computed as the cluster data point means.

The input parameters of LAC are the number of clusters K and h. The latter is used to control how the data point variance along a specific dimension translates to the weight. In addition to these parameters, the clustering

produced by LAC depends heavily on the initial choice of the centroids. LAC does not contain a method for outlier handling.

2.3.8 Information-Theoretic Co-Clustering

The authors of Information-Theoretic Co-Clustering [18] consider data matrices that can be interpreted as contingency tables; examples include word-document co-occurrence tables and webpage-user browsing statistics. A given contingency table is treated as a joint probability distribution between two discrete random variables that take values over the rows and columns.

Let us write X and Y for the random variables taking values in the set of rows and the set of columns, respectively. Also, let us write $\hat{X}$ and $\hat{Y}$ for the random variables taking values in the set of row clusters and the set of column clusters, respectively. The *mutual information* $I(X, Y)$ between two random variables is a measure of the amount of information that they contain about each other. The optimal co-clustering is defined as the clustering that minimizes the value of the objective function $I(X, Y) - I(\hat{X}, \hat{Y})$, given the number of the row and the column clusters. The authors present an iterative algorithm that monotonically decreases the value of the objective function.

The only input parameters required by the algorithm are the number of row and column clusters. The authors show good experimental results on clustering word-document co-occurrence data are and argue that their algorithm works especially well on sparse high-dimensional data. The authors have later extended their work to a more general co-clustering framework [9].

It is important to note that co-clustering as defined here and subspace clustering have different application areas. Consider a data matrix with a given co-clustering. The value of the cost function of the Information-Theoretic Co-Clustering algorithm is optimal if all co-cluster blocks have constant values (zero variance within the co-clusters). However, most subspace clustering algorithms could not identify any subspaces based on a constant-valued block matrix. The row clusters would be found, but all dimensions of the data space would be considered relevant, since the within-cluster variance on each dimension would be zero.

Chapter 3

Cluster Validation

After we have run a clustering algorithm on a data set, we have to be able to evaluate the quality of the resulting clustering. Does the clustering make sense? Is the result reliable? Does the algorithm produce similar clusterings on different runs? Is the result close to the true clustering (if there is one)? Evaluating the quality of a clustering is referred to as *cluster validation*. Cluster validation is useful for instance in selecting the number of clusters or other parameters of the algorithms, or in comparing clustering algorithms. [30, 56]

In this section, we present an overview of the most popular methods for cluster validation. They can be divided into two categories: *external cluster validation* and *internal cluster validation*.[1] We will see that external cluster validation is based on calculating distances between clusterings, and many internal cluster validation methods require cluster comparison methods as well. This has motivated us to develop clustering distances for subspace clusterings.

Recall that a *clustering* $\mathcal{C}$ is a partitioning of the set of m data points into disjoint clusters $C_1, C_2, \ldots, C_K$ of sizes $m_1, m_2, \ldots, m_K$, where $\sum_i m_i = m$. We will refer to this kind of clusterings as *ordinary clusterings* whenever we want to emphasize the difference from subspace clusterings.

3.1 External Cluster Validation

If we test a clustering algorithm on a synthetic data set, we usually know the true clustering for the data. In these cases, we would like to be able to calculate the distance between the true clustering and the clustering produced by our algorithm. If we knew which clusterings are close to the true clustering, we would be able to choose among parameter values or different clustering

[1] Different divisions are possible: [56] suggests external, internal, and relative cluster validation.

algorithms. Evaluating the quality of a clustering by comparing it to a true clustering is referred to as external cluster validation. There are numerous methods for comparing ordinary clusterings: In this section, we present the Clustering Error measure, several methods based on point pair counting, the Variation of Information measure, and Hubert's modified Γ statistic.

Virtually all criteria for comparing clusterings can be computed given the so-called *confusion matrix*. Assume that we have two clusterings $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ and $\mathcal{C}' = \{C'_1, C'_2, \ldots, C'_{K'}\}$. The confusion matrix $M = (m_{ij})$ is a $K \times K'$ matrix whose ijth element is the number of points in the intersection of clusters C_i and C'_j, i.e., $m_{ij} = |C_i \cap C'_j|$.

3.1.1 Clustering Error

An intuitive way to compare clusterings is to calculate the *Clustering Error* (CE). It is the proportion of points which are clustered differently in $\mathcal{C}$ and $\mathcal{C}'$ after an optimal matching of clusters. (If $K \neq K'$, we simply create $|K - K'|$ empty clusters to match the remaining clusters.) In other words, it is the scaled sum of the non-diagonal elements of the confusion matrix, minimized over all possible permutations of rows and columns (permutations of the cluster labels). In practice, we do not need to try out all possible permutations; the Clustering Error can be computed in $O(K^3)$ with the help of the Hungarian method, presented as Algorithm 1 [29, 46].[2] The Clustering Error is a metric.

3.1.2 Measures Based on Counting Point Pairs

An important class of criteria for comparing clusterings is based on counting the pairs of points on which two clusterings agree/disagree. Each pair of data points falls in one of the four categories labeled as N_{11}, N_{10}, N_{01}, and N_{00}. The category N_{11} contains the pairs of points that are in the same cluster both in $\mathcal{C}$ and in $\mathcal{C}'$. The category N_{10} contains the pairs of points that are in the same cluster in $\mathcal{C}$ but not in $\mathcal{C}'$. The definitions of N_{01} and N_{00} are similar. All four

[2]Note that we have presented the Hungarian method in its classical cost minimization form: conversion into a cost maximization problem can be done simply by multiplying the matrix elements by -1. Also, in the algorithm description, we have used the more general term 'cost matrix' instead of 'confusion matrix'. The Hungarian method is used to modify the cost matrix in such a way that the optimal permutation of the rows/columns is easy to find.

[3]This step is not trivial; the rather long algorithm is presented in [29].

Algorithm 1. Hungarian method.
Input: A cost matrix M of size $K \times K'$.
Output: A modified cost matrix M' of size $\max(K, K') \times \max(K, K')$ in which it is possible to find one or more permutations of the rows/columns such that the total cost (the sum of the diagonal elements) becomes zero. The same permutations (excluding the extra rows/columns) are optimal also for the original cost matrix M.

1. Make the matrix square by adding rows or columns of zeroes if necessary. The matrix is now of size $\max(K, K') \times \max(K, K')$.
2. Subtract the row minimum from the entries of each row. Each row now has at least one zero.
3. Subtract the column minimum from the entries of each column. Each row and each column now has at least one zero.
4. Select rows and columns across which you draw lines, in such a way that all the zeroes are covered and that no more lines have been drawn than necessary.[3]
5. (i) If the number of the lines is $\max(K, K')$, return the modified cost matrix.
   (ii) If the number of the lines is smaller than $\max(K, K')$, go to step 6.
6. Find the smallest element which is not covered by any of the lines. Then subtract it from each entry which is not covered by the lines and add it to each entry which is covered by both a vertical and a horizontal line. Go back to step 4.

counts can be obtained from the confusion matrix:

$$N_{11} = \frac{1}{2}\left[\sum_{i,j} m_{ij}^2 - m\right],$$
$$N_{10} = \sum_{i=1}^{K'}\sum_{j=1}^{K}\sum_{k=j+1}^{K} m_{ij}m_{ik},$$
$$N_{01} = \sum_{i=1}^{K}\sum_{j=1}^{K'}\sum_{k=j+1}^{K'} m_{ji}m_{ki},$$
$$N_{00} = N - N_{11} - N_{01} - N_{10}.$$

Above $N = m(m-1)/2$ is the total number of point pairs.

An example of point-pair counting based methods are the two Wallace indices [62]. The first Wallace index represents $\mathcal{W}_I$ the probability that a pair of points which is in the same cluster in $\mathcal{C}$ is in the same cluster also in $\mathcal{C}'$. That is,

$$\mathcal{W}_I(\mathcal{C},\mathcal{C}') = \frac{N_{11}}{\sum_k m_k(m_k-1)/2}, \tag{3.1}$$

where m_k is the number of data points in the cluster C_k. The second Wallace index is defined similarly, flipping the two clusterings:

$$\mathcal{W}_{II}(\mathcal{C},\mathcal{C}') = \frac{N_{11}}{\sum_{k'} m'_{k'}(m'_{k'}-1)/2}. \tag{3.2}$$

Here, m'_k is the number of data points in the cluster C'_k. The Wallace indices are asymmetric; a symmetric alternative is the Fowlkes-Mallows index $\mathcal{F}$ [25], which is just the geometric mean of the two Wallace indices:

$$\mathcal{F}(\mathcal{C},\mathcal{C}') = \sqrt{\mathcal{W}_I(\mathcal{C},\mathcal{C}')\mathcal{W}_{II}(\mathcal{C},\mathcal{C}')}. \tag{3.3}$$

The Rand index [52] is the proportion of point pairs on which the two clusterings agree, given as

$$\mathrm{Rand}(\mathcal{C},\mathcal{C}') = \frac{N_{11}+N_{00}}{N}, \tag{3.4}$$

where $N = N_{11} + N_{10} + N_{01} + N_{00}$, the total number of pairs of points. The Mirkin metric [43] $\mathcal{K}$ is closely related:

$$\mathcal{K}(\mathcal{C},\mathcal{C}') = 2(N_{01}+N_{10}) = m(m-1)(1-\mathrm{Rand}(\mathcal{C},\mathcal{C}')). \tag{3.5}$$

One of the problems of the Rand index is that as the number of clusters grows, the quantity N_{00} starts to dominate. The Jaccard index [40] $\mathcal{J}$ fixes

this problem by subtracting N_{00} from both the nominator and denominator of the Rand index, and is thus given as

$$\mathcal{J}(\mathcal{C}, \mathcal{C}') = \frac{N_{11}}{N_{11} + N_{01} + N_{10}}. \tag{3.6}$$

Out of the methods mentioned, the quantity (1 - Rand index) and the Mirkin measure are metrics. Some of these measures have been empirically compared in [42]: It has been observed that the Rand, the Jaccard, and the Fowlkes-Mallows index are generally consistent with each other when the number of clusters, the cluster size, and the dimensionality of the data are varied.

3.1.3 Variation of Information

The *Variation of Information (VI)* [40] is a recently proposed clustering distance based on information theoretic concepts; let us start by introducing these. The *entropy* of a clustering $\mathcal{C}$ measures the uncertainty related to the cluster membership of a random data point. It is defined as

$$\mathrm{H}(\mathcal{C}) = -\sum_{i=1}^{K} p(i) \log p(i), \tag{3.7}$$

where $p(i) = m_i/m$ is the proportion of the points in cluster C_i. The *conditional entropy* $\mathrm{H}(\mathcal{C}|\mathcal{C}')$ is the remaining entropy in clustering $\mathcal{C}$ when clustering $\mathcal{C}'$ is known, i.e.,

$$\mathrm{H}(\mathcal{C}|\mathcal{C}') = -\sum_{i=1}^{K}\sum_{j=1}^{K'} p(i,j) \log p(i|j). \tag{3.8}$$

In this expression, $p(i,j) = m_{ij}/m$ and $p(i|j) = m_{ij}/m_j$. The VI distance of two clusterings is simply the sum of two conditional entropies,

$$\mathrm{VI}(\mathcal{C}, \mathcal{C}') = \mathrm{H}(\mathcal{C}|\mathcal{C}') + \mathrm{H}(\mathcal{C}'|\mathcal{C}). \tag{3.9}$$

An equivalent way of writing the VI distance is

$$\mathrm{VI}(\mathcal{C}, \mathcal{C}') = 2\mathrm{H}(\mathcal{C}, \mathcal{C}') - \mathrm{H}(\mathcal{C}) - \mathrm{H}(\mathcal{C}'), \tag{3.10}$$

where $\mathrm{H}(\mathcal{C}, \mathcal{C}')$ is the *joint entropy* of the two clusterings, defined as

$$\mathrm{H}(\mathcal{C}, \mathcal{C}') = -\sum_{i=1}^{K}\sum_{j=1}^{K'} p(i,j) \log p(i,j). \tag{3.11}$$

Eq. 3.10 leads to the simple expression

$$\begin{aligned} \mathrm{VI}(\mathcal{C},\mathcal{C}') &= \sum_{i=1}^{K}\sum_{j=1}^{K'} p(i,j)\log\frac{p(i)p(j)}{p^2(i,j)} \\ &= \frac{1}{m}\sum_{i=1}^{K}\sum_{j=1}^{K'} m_{ij}\log\frac{m_i m'_j}{m_{ij}^2}. \end{aligned} \tag{3.12}$$

Note that also the VI distance can be computed based only on the confusion matrix.

3.1.4 Hubert's Γ Statistic

Consider two clusterings $\mathcal{C}$ and $\mathcal{C}'$. Define a $m \times m$ matrix X as follows.[4] Set $X_{ij} = 1$ if the data points x_i and x_j belong to the same cluster in C, and 0 otherwise. Define the matrix Y similarly but using clustering C'. If the matrices X and Y are similar, so are the two clusterings. One way to compare X and Y is to compute *Hubert's normalized* Γ *statistic*, or the correlation of the elements in the two data matrices [56]. It is given as

$$\Gamma(X,Y) = \frac{(1/N)\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}(X_{ij}-\mu_X)(Y_{ij}-\mu_Y)}{\sigma_X\sigma_Y}. \tag{3.13}$$

Here μ_X, μ_Y, σ_X, and σ_Y are the means and the variances of the matrix elements, and $N = m(m-1)/2$. The values of $\Gamma(X,Y)$ are between -1 and 1; high values indicate similar clusterings. An equivalent way of expressing the Γ statistic is

$$\Gamma(X,Y) = \frac{NN_{11} - N_1N_2}{\sqrt{N_1N_2(N-N_1)(N-N_2)}}, \tag{3.14}$$

where $N_1 = N_{11} + N_{10}$, and $N_2 = N_{01} + N_{00}$. It can thus be computed using only the information in the confusion matrix.

3.1.5 Hypothesis Testing in External Cluster Validation

We have introduced several distance measures for clusterings. When we use these distance measures in external cluster validation, we face the problem of interpreting the result. If we compare a clustering to a true clustering, what does it mean to get a Rand index value 0.6 — is our clustering hopelessly bad or relatively good? Is there even any cluster structure in the data?

[4] Recall that m stands for the number of rows in the data matrix.

It is important to note that the distribution of the values of a given distance measure depends on the data set and the number of clusters. For instance, recall that the quantity N_{00} tends to dominate the Rand index especially if the number of clusters is high. Therefore, in practice, the Rand index rarely gives values close to zero — it has been observed that the useful range of the index is sometimes as narrow as $(0.6, 1]$ [40].

It is thus important to relate the index value to the whole distribution of the possible index values. Assuming that the null hypothesis holds (for instance, that the data does not possess any clustering structure), we want to calculate the probability of observing a given value of the index. This gives us a p-value that can be compared with the critical value of our statistical test. In practice, it is impossible to investigate the appropriate distribution analytically, and one has to resort to empirical estimates.

In generating the empirical distribution, we have to make some assumptions. For instance, the *random label hypothesis* assumes that all possible labelings of the data are equally likely, given the number of clusters K and fixed cluster sizes. The *random position hypothesis* assumes that all arrangements of the m data points in a specific region of the space are equally likely (usually the same region in which the original data points lie). The design of the hypothesis test requires care; more information on the topic can be found in [56].

Distribution considerations are useful also for normalizing the range of the external cluster validation measures. For each external cluster validation measure q, there exists a corresponding 'corrected' measure q'. It is a normalized version of q, given as

$$q' = \frac{q - \mathrm{E}(q)}{\max q - E(q)}, \tag{3.15}$$

where $\max q$ is the maximum possible value of q and $\mathrm{E}(q)$ is the expectation of q under the null hypothesis [56]. After the normalization, the expected value of the index is 0, while the maximum (attained for identical clusterings) is 1. The null hypothesis could be for instance that the two clusterings are sampled independently from the set of all partition pairs with a fixed number of points in each cluster. Of course, these assumptions are not entirely realistic [40]. As an example of index normalization, the corrected version of the Rand index is in wide use; the somewhat complicated expression is given in [42].

3.1.6 External Cluster Validation for Subspace Clusterings

We are not aware of any existing methods for calculating distances between subspace clusterings, which makes external cluster validation for subspace clus-

terings impossible. Subspace clusterings differ so much from ordinary clusterings that the methods presented above cannot directly be used to compare them.

For instance, consider the simple case of axis-aligned subspace clusterings, where a cluster can be viewed as a collection of data matrix elements. In general, not all matrix elements belong to a subspace cluster, and further, a given matrix element may be part of several subspace clusters. The methods presented above require the clusters to form a partition of the data, and therefore cannot be used to compare axis-aligned subspace clusterings.

In some cases we could compare subspace clusterings by comparing only the corresponding row clusterings. However, leaving the subspaces out from the considerations would not produce a fair result. Moreover, in the general case, the row clusters do not form a partition of the whole set of data points, and again, the traditional methods for comparing clusterings cannot be applied.

Already the case of axis-aligned subspace clusterings is problematic, and non-axis aligned subspace clusterings differ even more from traditional clusterings. We therefore need to develop new methods if we want to calculate distances between subspace clusterings.

3.2 Internal Cluster Validation

Evaluating the quality of a clustering without any knowledge of a true clustering is referred to as internal cluster validation. Internal cluster validation methods can be roughly divided into two classes. The *point configuration based methods* assume a model for a high-quality clustering — for instance, a clustering with well-separated compact clusters might be considered good — and judge a given clustering based on the model. These methods can therefore be used only with a subset of clustering algorithms; for example, density-based methods such as DBSCAN [24] are excluded, since these methods produce clusters of varying shapes.

The *stability based methods* are applicable more generally; they do not assume anything about the point configuration of a good clustering. The underlying idea is that a stable clustering is a good clustering. Commonly, a given clustering algorithm is run several times on samples of a data set, and if the the clustering results on different samples are similar[5], it is concluded that the clustering is stable and therefore of good quality.

Naturally, the stability based methods cannot be applied if we do not have a distance measure for clusterings. On the other hand, distance measures

[5] As we will see later, one way to define sample similarity here is to compare the clusterings at the intersection of the samples

are not needed with the point configuration based methods. However, as we will see in Section 3.2.3, the point configuration based methods are not easily generalizable to subspace clusterings, and at present, the stability based methods are our only option for doing internal cluster validation for subspace clusterings.

Next, we present an overview of the most common internal cluster validation methods.

3.2.1 Point Configuration Based Methods

The Dunn index. There are several possible ways to define the dissimilarity between the clusters C_i and C_j; the definition used with the Dunn index is

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y), \tag{3.16}$$

where $x, y \in \mathbb{R}^p$ are data vectors, and $d(x, y)$ is a distance between x and y. Let us also define the diameter of a cluster C_i as

$$\text{diam}(C_i) = \max_{x,y \in C_i} d(x, y). \tag{3.17}$$

The diameter is a measure of the dispersion of a cluster. Now, the Dunn index is given as

$$\mathcal{D}(\mathcal{C}) = \min_{1 \leq i \leq K} \min_{1 \leq j \leq K, j \neq i} \frac{d(C_i, C_j)}{\max_{1 \leq k \leq K} \text{diam}(C_k)}. \tag{3.18}$$

The Dunn index will be large for clusterings with compact and well-separated clusters.

The Davies-Bouldin index. The Davies-Bouldin index assumes that a good clustering comprises of well-separated, compact, spherical clusters. If $w_i \in \mathbb{R}^p$ and $w_j \in \mathbb{R}^p$ are the centroids of the clusters C_i and C_j, respectively, we can define the dissimilarity d_{ij} between these two clusters as $d_{ij} = ||w_i - w_j||_r$, the L_r-norm of the difference between the centroid vectors. Also, let us define the dispersion of a cluster C_i as

$$s_i = \left(\frac{1}{m_i} \sum_{x \in C_i} ||x - w_i||^r \right)^{1/r}, \tag{3.19}$$

where m_i is the number of data points in the ith cluster, and $r \in \mathbb{Z}_+$. Define a similarity index R_{ij} for the two clusters as

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}, \tag{3.20}$$

This quantity can be shown to satisfy a set of intuitive conditions [17]. Now, the Davies-Bouldin index is given as

$$\mathcal{DB}(\mathcal{C}) = \frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i} R_{ij}. \tag{3.21}$$

Because it is desirable for the clusters to be as dissimilar as possible, we prefer small values of the Davies-Bouldin index. [17, 56]

The modified Hubert Γ statistic. Let us define a $m \times m$ matrix Q, whose element $Q_{ij} = d(w_{k_i}, w_{k_j})$, where w_k is the centroid of the kth cluster and k_i is the index of the cluster to which the data point x_i belongs. Also, let D denote the data distance matrix: $D_{ij} = d(x_i, x_j)$ is just the distance between the ith and the jth data point. We can compare the matrix Q with the data distance matrix D using Hubert's Γ statistic from Eq. 3.13. Essentially, we compute the correlation of the data point distances with the distances of the corresponding cluster centroids. When $d(w_{k_i}, w_{k_j})$ is close to $d(x_i, x_j)$ for all i, j, the clusters are compact and the value of Γ will be high. [56]

The Figure of Merit. The Figure of Merit [64] is also based on the assumption that the data points form compact, spherical clusters. The idea is to apply the clustering algorithm to the data set without one of the original attributes. The remaining attribute is used to assess the predictive power of the resulting clusters: meaningful clusters should exhibit less variation in the remaining attribute than clusters formed by coincidence. The Figure of Merit (FOM) is defined as the average squared distance of the attribute values from their mean in each cluster, for the attribute that was not used in the clustering. The Aggregate Figure of Merit is computed by leaving out each attribute in turn and calculating the average of the resulting FOM scores.

Other measures and a comparison. Several extensions of the Dunn index and the Davies-Bouldin index exist [22, 11], as well as numerous other measures based on similar ideas, including the Silhouette index, the C index, and the Goodman-Kruskal index [56]. Some of these indices have been compared in [22, 8, 11].[6]

3.2.2 Stability Based Methods

The stability based methods can be roughly divided into methods that repeatedly split the data set into two parts and methods that sample the original data set. Our discussion follows an excellent review of the topic presented in [37].

[6]Since point configuration based internal cluster validation is not in the main focus of this thesis, we have decided to leave out the more complicated extensions from our overview.

Splitting the Data Set into Two

The stability measure of Lange et al. *The stability measure of Lange et al.* [37] is based on splitting the data set into two halves and evaluating the agreement of the clusterings on these two halves of the data.

Let us write $A_K(X)$ for the result of a clustering algorithm A that produces K clusters on a data set X. The data set is split into two halves X' and X''; both are clustered using the algorithm A, resulting in clusterings $A_K(X')$ and $A_K(X'')$. Since we have clustered disjoint sets of data points, we compare the clusterings with the help of a classifier. In order to evaluate the agreement of the two clusterings, the cluster labels of $A_K(X')$ are used to train a classifier ϕ (for instance a K-nearest-neighbor classifier) in the data space $\mathbb{R}^p$. This classifier can now be used to classify the points of X'', resulting in classification $\phi(X'')$. Finally, $A_K(X'')$ and $\phi(X'')$ are compared using the Clustering Error measure. Good agreement and a small value of CE are desirable, since they imply a stable clustering.

In practice, this data set splitting procedure is repeated r times ($r = 20$ in the original experiments [37]), and the average CE is used as the algorithm stability score $S(A_K)$. Different classifiers are experimented with, and the one giving the best CE values is selected. Since the CE values are affected by the number of clusters K, the stability score is normalized by $S(R_K)$, the so-called misclassification rate of random labelings with K clusters. This is calculated as the mean CE of r pairs of random labelings. The final stability score is given as $S(A_K)/S(R_K)$; the minimum score determines the optimal number of clusters.

Clest. *Clest* [23] is very similar to the stability measure of Lange et al. It also splits the data set into two parts (the sizes of which can be determined by the user), clusters both parts, trains a classifier on one of the parts (the classifier can be decided by the user), uses the classifier to classify the second part, and compares this classification with the original clustering on the second part (the clustering distance measure is also for the user to choose; the Fowlkes-Mallows index is used in the original experiments). The data is split r times, and since the result depends on K, the median similarity measure is compared to a corresponding measure for data sets drawn from a null reference distribution. Clest contains a large number of input parameters, even more than the ones mentioned here, and the user is not given much guidance on how to choose good values for the parameters.

Prediction Strength. *The Prediction Strength* measure [57] is calculated almost exactly as the stability measure of Lange et al. and Clest. The classifier used here is the nearest class centroid classifier, which brings in the assumption of spherical clusters. The similarity of clustering solutions is assessed by

essentially measuring the intersection of the two clusters that match worst. A user-specified threshold on the average similarity score is used to estimate the number of clusters.

Sampling the Data

Model Explorer. *The Model Explorer Algorithm* [10] repeatedly clusters two non-disjoint samples of the data set. First, two samples of equal size are generated (their size is 80% of the data points in the experiments) and both of them are clustered. The clusterings are then compared for the data points that belong to both samples. The clustering distance measure is a parameter of the method; the Fowlkes-Mallows index is used in the original experiments. A pair of samples is generated r times, and a "jump" in the mean clustering distance graph is used to determine a good number of clusters.[7]

The Re-sampling Approach by Levine and Domany. *The Re-sampling Approach by Levine and Domany* [38] creates r samples of the data and clusters both the samples and the original data set. The authors define a figure of merit M[8] that measures the extent to which the clusterings on the samples are in agreement with the clustering on the full data set. The authors suggest selecting the optimal number of clusters based on the local maxima of M, but this is often difficult, since several local maxima can occur.

Comparing Stability-Based Methods

Lange et al. [37] present an experimental comparison of 6 internal cluster validation measures (5 of the stability measures presented above and the Gap statistic [58]) on both synthetic and real data sets. The measures are used to decide the optimal number of clusters for each data set. Surprisingly, the results are not encouraging. The predictions of the various measures vary wildly on most data sets, also on artificial ones; the practitioner must exercise caution in choosing a method to use.

The numerous free parameters of the stability based methods make them somewhat difficult to use. Further, the techniques for choosing the number of clusters are often not very well justified. The stability measure by Lange et al. [37] is a welcome exception in these respects. Perhaps the most serious problem of the stability-based methods is that they cannot take in account the effect of inherent non-determinism of many clustering algorithms. If the clustering result depends on the initialization of the algorithm, the algorithm

[7]This can be compared to choosing a good dimensionality of a subspace in principal component analysis based on a "jump" in a graph of the cumulative sum of eigenvalues.

[8]Note that the figure of merit of Levine and Domany is different from the Figure of Merit presented in Section 3.2.1.

would not be judged stable even if it was run several times on the whole data set, let alone on samples. Perhaps a combination of the stability-based techniques and meta-clustering could provide a solution to this problem.

3.2.3 Internal Cluster Validation for Subspace Clusterings

The point configuration based methods are not directly applicable to subspace clusterings. Usually, to be able to use these methods, we need to be able to calculate two things: a distance between two data vectors in the same cluster, and a distance between two clusters. In the case of subspace clusterings, calculating these is problematic. If x_i, x_j belong to one cluster and x_k, x_l in another cluster, the distances $d(x_i, x_j)$ and $d(x_k, x_l)$ have to be calculated in the appropriate subspaces, and it is not clear how to scale the distances with the dimensionality of these subspaces. Also, if we want to calculate the distance $d(w_1, w_2)$ between the centroids of two subspace clusters C_1 and C_2, it is unclear which subspace we should use. We cannot calculate this distance in the full space, since the contribution of the irrelevant dimensions is essentially random, and it might distort the distance considerably, especially in the case of low-dimensional subspace clusters. In the special case in which the subspaces related to the two clusters are orthogonal, it does not even make sense to calculate the distance between the centroids.

The stability based methods can be applied to subspace clusterings just as easily as to ordinary clusterings. However, in the case of subspace clusterings, we have the choice of sampling only the rows or both the rows and the columns of the data matrix.

Chapter 4

Comparing Axis-Aligned Subspace Clusterings

For the sake of simplicity, we start by considering axis-aligned subspace clusterings, the simplest and most popular type of subspace clusterings. We extend our analysis to non-axis-aligned subspace clusterings and other related types of clusterings in Chapter 5.

4.1 Size, Union, and Intersection of Axis-Aligned Subspace Clusters

Our data matrix $X = (x_{ij})$ has m rows and p columns. Recall that an *axis-aligned subspace cluster* S is a pair (R, C), where $R \subseteq \{r_1, r_2, \ldots, r_m\}$ is a subset of the rows and $C \subseteq \{c_1, c_2, \ldots, c_p\}$ is a subset of the columns. A *axis-aligned subspace clustering* $\mathcal{S}$ is a collection $\{S_1, S_2, \ldots, S_K\}$ of K subspace clusters.

In order to construct a clustering distance measure for axis-aligned subspace clusterings, we need to define the size, the union, and the intersection of subspace clusters and clusterings. To this end, we define the *support* of a cluster S_k as the set of matrix elements in it, given as $\text{supp}\,(S_k) = \{x_{ij} | r_i \in R_k \wedge c_j \in C_k\}$. The support of a clustering $\mathcal{S}$ is $\text{supp}\,(\mathcal{S}) = \bigcup_k \text{supp}\,(S_k)$. The size $|S_k|$ of a cluster is the number of matrix elements in its support. Similarly, the size $|\mathcal{S}|$ of a clustering is the number of matrix elements in its support. The union and the intersection of two subspace clusters are given as the union and the intersection of their supports. We denote the union of two subspace clusterings $\mathcal{S}$ and $\mathcal{S}'$ by $U = U(\mathcal{S}, \mathcal{S}') = \text{supp}\,(\mathcal{S}) \cup \text{supp}\,(\mathcal{S}')$ and the intersection by $I = I(\mathcal{S}, \mathcal{S}') = \text{supp}\,(\mathcal{S}) \cap \text{supp}\,(\mathcal{S}')$.

To simplify the analysis, we will start by considering only *disjoint subspace*

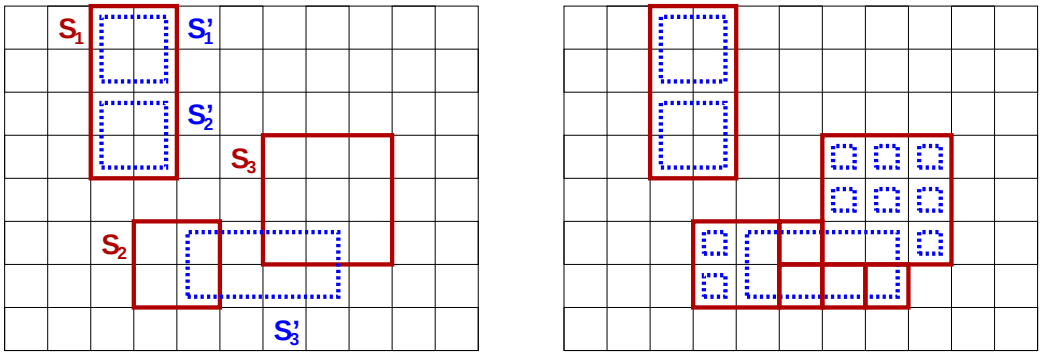


Figure 4.1: (Left) Two axis-aligned subspace clusterings which we wish to compare. The solid rectangles depict the clustering $\mathcal{S} = \{S_1, S_2, S_3\}$, and the dashed rectangles the clustering $\mathcal{S}' = \{S'_1, S'_2, S'_3\}$. For these clusterings, $\mathrm{CE}(\mathcal{S}, \mathcal{S}') = 19/25$ and $\mathrm{RNIA}(\mathcal{S}, \mathcal{S}') = 13/25$. (Right) To create a partition of the matrix elements, we have filled the non-intersecting areas with singleton clusters $S_4, \ldots, S_7$ and $S'_4, \ldots, S'_{12}$. This allows us to compute $\mathrm{VI}(\mathcal{S}, \mathcal{S}') = 1.68$ and $1 - \mathrm{Rand}(\mathcal{S}, \mathcal{S}') = 82/300$.

clusterings, i.e., clusterings in which the clusters are disjoint in the sense that they do not share matrix elements: $\mathrm{supp}\,(S_k) \cap \mathrm{supp}\,(S_l) = \emptyset$ for all $k, l \in \{1, \ldots, K\}$ with $k \neq l$. It is important to note that any given data point or an attribute may still be relevant for subspaces of multiple clusters. A large number of subspace clustering algorithms are guaranteed to produce disjoint clusterings; exceptions include algorithms of the type of CLIQUE [4] and SUBCLU [33]. We discuss the extension of the analysis to non-disjoint subspace clusterings in Section 4.4.

4.2 Properties of a Distance Measure

Comparing subspace clusterings differs greatly from comparing ordinary clusterings, since subspace clusterings are not partitions of the data matrix. We will therefore start by introducing a set of properties for describing a subspace clustering comparison distance $d(\mathcal{S}, \mathcal{S}')$. We will use these properties to characterize and compare different distance measure candidates.

Metric. A distance measure d is a *metric* if it satisfies three axioms: positivity, symmetricity, and triangle inequality.

Positive. For all $\mathcal{S}$ and $\mathcal{S}'$ we have $d(\mathcal{S}, \mathcal{S}') \geq 0$ and $d(\mathcal{S}, \mathcal{S}') = 0$ if and only if $\mathcal{S} = \mathcal{S}'$.

Symmetric. For all $\mathcal{S}$ and $\mathcal{S}'$ we have $d(\mathcal{S}, \mathcal{S}') = d(\mathcal{S}', \mathcal{S})$.

Triangle inequality. For all $\mathcal{S}$, $\mathcal{S}'$, and $\mathcal{S}''$ we have $d(\mathcal{S}, \mathcal{S}') \leq d(\mathcal{S}, \mathcal{S}'') + d(\mathcal{S}'', \mathcal{S}')$.

Label permutation invariant. An alternative way to consider an axis-aligned subspace clustering $\mathcal{S} = \{S_1, S_2, \ldots, S_K\}$ is to view it as a function $\mathcal{S} : \{x_{ij}\} \to \{1, 2, \ldots, K\}$. This function assigns cluster indices to data matrix elements: we refer to the cluster index $\mathcal{S}(x_{ij})$ as the *cluster label* of the element x_{ij}. A *permutation* $\rho : \{1, 2, \ldots, K\} \to \{1, 2, \ldots, K\}$ is a bijection from the set of cluster labels to the set of cluster labels. Let us permute the cluster labels of $\mathcal{S}$ by ρ and denote the resulting clustering by $\mathcal{S}^\rho$. In other words, we have $\mathcal{S}^\rho(x_{ij}) = \rho(\mathcal{S}(x_{ij}))$. A clustering distance measure d is *label permutation invariant* if $d(\mathcal{S}^\rho, \mathcal{S}'^\pi) = d(\mathcal{S}, \mathcal{S}')$ for any permutations ρ and π and any clusterings $\mathcal{S}$ and $\mathcal{S}'$.

Penalty for non-intersecting area. Consider two clusterings $\mathcal{S}$ and $\mathcal{S}'$, the data matrix elements in their union U, in their intersection I, and in $U \backslash I$. Let us refer to $|U \backslash I|$ as the *non-intersecting area* of these two clusterings. Consider adding one or more unclustered data matrix elements $x_{ij} \notin U$ to $U \backslash I$. In effect, we increase the non-intersecting area of the two clusterings while keeping everything else unchanged. The new matrix elements might have been added to $\mathcal{S}$ only, $\mathcal{S}'$ only, or to both clusterings. Let us denote the new clusterings by $\mathcal{S}^U$ and $\mathcal{S}'^U$; note that one of these clusterings might in fact be equal to the original one. A distance measure d *penalizes for non-intersecting area* if $d(\mathcal{S}^U, \mathcal{S}'^U) > d(\mathcal{S}, \mathcal{S}')$ for any clusterings $\mathcal{S}$ and $\mathcal{S}'$.

Background independent. Consider two clusterings $\mathcal{S}$ and $\mathcal{S}'$ on the data matrix X of size $m \times p$. Let us introduce an alternative notation $\mathcal{S}_X$ and $\mathcal{S}'_X$ in order to emphasize that we have clusterings on X. Now consider adding m' rows and p' columns to X and denote the new $(m + m') \times (p + p')$ data matrix by X'. The matrix element sets $\mathrm{supp}\,(\mathcal{S})$ and $\mathrm{supp}\,(\mathcal{S}')$ are included in X', so we can write $\mathcal{S}_{X'}$ and $\mathcal{S}'_{X'}$ for the same clusterings on the larger data matrix X'. A distance measure d is *background independent* if $d(\mathcal{S}_X, \mathcal{S}'_X) = d(\mathcal{S}_{X'}, \mathcal{S}'_{X'})$ for any clusterings $\mathcal{S}$ and $\mathcal{S}'$ and $m' > 0$ or $p' > 0$.[1]

[1] To motivate the background independence property, let us consider the following. The distance $d(\mathcal{S}, \mathcal{S}')$ should not be affected by the size of the data matrix X, only the size of the union U of the two clusterings. For instance, increasing the size of X by adding noise rows and columns should not move the clusterings $\mathcal{S}$ and $\mathcal{S}'$ closer to each other.

Scale invariant. We define scaling the data matrix X by a constant $k \in \mathbb{Z}^+$ as introducing k copies of each row and column of the matrix. Let us denote the scaled data matrix by kX. While X has rows $\{r_1, \ldots, r_m\}$ and columns $\{c_1, \ldots, c_p\}$, kX has rows $\{r_{11}, r_{12}, \ldots, r_{1k}, \ldots, r_{m1}, r_{m2}, \ldots, r_{mk}\}$ and columns $\{c_{11}, r_{12}, \ldots, c_{1k}, \ldots, c_{m1}, c_{m2}, \ldots, c_{mk}\}$. Now consider a subspace clustering $\mathcal{S}$ on X and its scaled version $k\mathcal{S}$ on kX. If a cluster $S_i = (R_i, C_i)$ of $\mathcal{S}$ has the row r_j in its row set R_i, then the cluster $kS_i = (kR_i, kC_i)$ of $k\mathcal{S}$ has the rows $r_{j1}, r_{j2}, \ldots, r_{jk}$ in its row set kR_i. Similarly, if a cluster $S_i = (R_i, C_i)$ of $\mathcal{S}$ has the column c_j in its column set C_i, then the cluster $kS_i = (kR_i, kC_i)$ of $k\mathcal{S}$ has the columns $c_{j1}, c_{j2}, \ldots, c_{jk}$ in its column set kC_i. A distance measure d is *scale invariant* if $d(k\mathcal{S}, k\mathcal{S}') = d(\mathcal{S}, \mathcal{S}')$ for all clusterings $\mathcal{S}$ and $\mathcal{S}'$ and for all $k \in \mathbb{Z}^+$.

Copy invariant. Consider two subspace clusterings $(\mathcal{S}, \mathcal{S}')$. Next consider introducing a disjoint copy of a this pair of clusterings in a large data matrix X, resulting in a pair of 'double clusterings' $(\mathcal{S}^D, \mathcal{S}'^D)$. In other words, introduce a copy $\tilde{\mathcal{S}}$ of $\mathcal{S}$ and a copy $\tilde{\mathcal{S}}'$ of $\mathcal{S}'$ such that the new cluster sizes $\{\tilde{m}_i\}$, $\{\tilde{m}'_i\}$ and cluster intersection sizes $\{\tilde{m}_{ij}\}$ equal to the old ones ($\{m_i\}$,$\{m'_i\}$,$\{m_{ij}\}$) and that $\text{supp}\,(\mathcal{S}) \cap \text{supp}\,(\tilde{\mathcal{S}}) = \emptyset$ and $\text{supp}\,(\mathcal{S}') \cap \text{supp}\,(\tilde{\mathcal{S}}') = \emptyset$. Then the 'double clusterings' are given by $\mathcal{S}^D = \{S_1, S_2, \ldots, S_K, \tilde{S}_1, \tilde{S}_2, \ldots, \tilde{S}_K\}$ and $\mathcal{S}'^D = \{S'_1, S'_2, \ldots, S'_{K'}, \tilde{S}'_1, \tilde{S}'_2, \ldots, \tilde{S}'_{K'}\}$ A distance measure d is *copy invariant* if $d(\mathcal{S}, \mathcal{S}') = d(\mathcal{S}^D, \mathcal{S}'^D)$ for all clusterings $\mathcal{S}$ and $\mathcal{S}'$.

Requires partitioning. Consider two clusterings $\mathcal{S}$ and $\mathcal{S}'$ and the set of matrix elements in their union U. In the general case, neither of these clusterings is a partition of U, since the elements in $\text{supp}\,(\mathcal{S})\backslash\text{supp}\,(\mathcal{S}')$ are not clustered by $\mathcal{S}'$, and the elements in $\text{supp}\,(\mathcal{S}')\backslash\text{supp}\,(\mathcal{S})$ are not clustered by $\mathcal{S}$. A subspace clustering distance measure d that *requires the clusterings to be partitions* of the data matrix elements in the union U therefore forces us to modify $\mathcal{S}$ and $\mathcal{S}'$ before we are able to compute $d(\mathcal{S}, \mathcal{S}')$. In order to transform $\mathcal{S}$ into a partition of the data matrix elements in U, we assign each $x_{ij} \notin \text{supp}\,(\mathcal{S}')\backslash\text{supp}\,(\mathcal{S})$ to a new singleton cluster. If $\mathcal{S}$ originally had K clusters, its modified version therefore has $K + |\text{supp}\,(\mathcal{S}')\backslash\text{supp}\,(\mathcal{S})|$ clusters. Similar transformation is done for $\mathcal{S}'$. See Fig. 4.1 for an example.

Multiple cluster coverage penalty. Consider two clusterings $\mathcal{S}$ and $\mathcal{S}'$ such that $\mathcal{S} = \{S_1\}$ consists only of a single cluster and $\mathcal{S}' = \{S'_1, \ldots, S'_{K'}\}$ consists of K' disjoint clusters of equal size with $\text{supp}\,(\mathcal{S}) = \text{supp}\,(\mathcal{S}')$. The clustering $\mathcal{S}'$ therefore clusters the same matrix elements than $\mathcal{S}$

but uses multiple clusters to cover the area. For an example of the case $K = 2$, see Fig. 4.1. A distance measure d *penalizes for multiple cluster coverage* if $d(\mathcal{S}, \mathcal{S}') \neq 0$ for $K' > 1$.

Generalizable. A distance measure for axis-aligned subspace clusterings is *generalizable* if it can also be applied to non-axis-aligned subspace clusterings and other related types of clusterings.

Handles ordinary clusterings. A distance measure $d(\mathcal{S}, \mathcal{S}')$ *handles ordinary clusterings* if it produces sensible results in the case in which $\mathcal{S}$ and $\mathcal{S}'$ are ordinary clusterings (partitions of the same element set).

Easy to compute.

Intuitive and understandable.

4.3 Distance Measures for Subspace Clusterings

We now present methods for comparing subspace clusterings by generalizing well-known distance measures for ordinary clusterings. We consider the Clustering Error (CE), the Rand index (as a well-known representative of the point pair counting based methods), and the Variation of Information (VI). In addition, we introduce a new distance measure, relative non-intersecting area (RNIA). We define and briefly discuss each of these four distance measures below; a comprehensive comparison of their properties is presented in Table 4.1.

Since we are comparing subspace clusterings, we consider the set of the data matrix elements $\{x_{ij}\}$ as our base element set, instead of the set of data points (rows).

4.3.1 Clustering Error

Consider subspace clusterings $\mathcal{S} = \{S_1, S_2, \ldots, S_K\}$ and $\mathcal{S}' = \{S'_1, S'_2, \ldots, S'_{K'}\}$ of K and K' clusters, respectively. Since the subspace clusterings are not partitions of the data matrix elements, we cannot form a confusion matrix M as in Section 2. Instead, let us define the *cluster intersection matrix* $T = (t_{ij})$ as a $K \times K'$ matrix in which t_{ij} is the number of data matrix elements shared by the clusters S_i and S'_j. More formally, $t_{ij} = |\mathrm{supp}\,(S_i) \cap \mathrm{supp}\,(S'_j)|$.

Let us transform T into a square matrix by adding rows or columns of zeroes if necessary and use the Hungarian method to find a permutation of the cluster labels such that the sum of the diagonal elements of T is maximized.

Denote this maximized sum by D_{max}. Now, we define the clustering error (CE) for subspace clusterings as

$$\mathrm{CE}(\mathcal{S}, \mathcal{S}') = \frac{|U| - D_{max}}{|U|}. \tag{4.1}$$

In the case of ordinary clusterings (partitions of the rows of the data matrix), the clustering error defined here is the clustering error of Section 2.

For the two clusterings of Fig. 4.1, the cluster intersection matrix T is

	S'_1	S'_2	S'_3
S_1	4	4	0
S_2	0	0	2
S_3	0	0	2

.

We also have $|U| = 25$, $D_{max} = 6$, and thus $\mathrm{CE}(\mathcal{S}, \mathcal{S}') = 19/25$.

4.3.2 Rand Index

The Rand index for ordinary clusterings is based on counting pairs of data points, but the Rand index for subspace clusterings is based on counting pairs of matrix elements. Recall that we need the quantities N_{11} (the number of pairs of matrix elements in the same cluster in both $\mathcal{S}$ and $\mathcal{S}'$), N_{00} (the number of pairs of matrix elements in a different cluster in both $\mathcal{S}$ and $\mathcal{S}'$) and N (the total number of pairs of matrix elements) for calculating the value of the Rand index according to Eq. 3.4.

Since we want our distance measure to be background independent, we will only count pairs of matrix element in U, the union of the two clusterings. Therefore $N = |U|(|U|-1)/2$. However, to compute the values of N_{01} and N_{10}, $\mathcal{S}$ and $\mathcal{S}'$ have to be partitions of U. To this end, we will make $\mathcal{S}$ a partition by filling the non-intersecting area $U \backslash \mathrm{supp}\,(\mathcal{S})$ with extra clusters. We will similarly convert $\mathcal{S}'$ into a partition of U.

If we considered this non-intersecting area area as a single big extra cluster, we would end up having zero distance between many different clusterings[2]. The only way around this difficulty seems to be to fill the non-intersecting area with extra singleton clusters, as illustrated in Fig. 4.1. After this filling procedure for both clusterings, we are able to compute the values of N_{11} and N_{00} as usual, for instance with the help of the confusion matrix.

When we discuss the Rand index, we actually consider $1 - \mathrm{Rand}(\mathcal{S}, \mathcal{S}') = (N_{01} + N_{10})/N$, since this is a proper distance measure: It assumes zero values for identical clusterings and positive values for non-identical clusterings.

[2]For instance, consider two single-cluster clusterings $\mathcal{S} = \{S_1\}$ and $\mathcal{S}' = \{S'_1\}$ such that $\mathrm{supp}\,(S_1) \cap \mathrm{supp}\,(S'_1) = \emptyset$.

If we transform the two clusterings of Fig. 4.1 into partitions by adding singleton clusters, the resulting confusion matrix M is

	S'_1	S'_2	S'_3	S'_4	S'_5	S'_6	...	S'_{12}	
S_1	4	4	0	0	0	0	...	0	8
S_2	0	0	2	1	1	0	...	0	4
S_3	0	0	2	0	0	1	...	1	9
S_4	0	0	1	0	0	0	...	0	1
S_5	0	0	1	0	0	0	...	0	1
S_6	0	0	1	0	0	0	...	0	1
S_7	0	0	1	0	0	0	...	0	1
	4	4	8	1	1	1	...	1	25

Based on this confusion matrix, we get $N = 300$, $N_{11} = 14$, $N_{00} = 204$, $N_{01} = 26$, $N_{10} = 56$, and $1 - \text{Rand}(\mathcal{S}, \mathcal{S}') = 82/300$.

4.3.3 Variation of Information

Like the Rand index, the Variation of Information (VI) also requires the clusterings to be partitions. For reasons similar to the case of the Rand index, we fill the non-intersecting areas of the clusterings with extra singleton clusters. After this step, we can easily compute the confusion matrix and thereby calculate the VI distance according to Eq. 3.12. For the two clusterings of Fig. 4.1, $\text{VI}(\mathcal{S}, \mathcal{S}') = 1.68$.

4.3.4 Relative Non-Intersecting Area

If we want to compare a subspace clustering $\mathcal{S}$ to a true clustering $\mathcal{S}'$, a simple approach would be to calculate the *precision*, the *recall*, and the *F-measure*, used widely in the information retrieval literature to measure the success of the retrieval task [53]. Retrieval is similar to subspace clustering in that it aims to extract a subset of the data that is alike in some respect, while the rest of the data is not assumed to be grouped in any way. Hence, a subspace clustering is like the unsupervised retrieval of several disjoint groups.

Using our subspace clustering notation, recall is defined as $|I|/\text{supp}\,(\mathcal{S}')$; it measures how big part of the matrix elements of the true clustering $\mathcal{S}'$ is retrieved (covered) by the clustering $\mathcal{S}$. Precision is defined as $|I|/\text{supp}\,(\mathcal{S})$; it measures the proportion of the matrix elements in the clustering $\mathcal{S}$ that belong to the true clustering $\mathcal{S}'$. The F-measure is just the geometric mean of the precision and the recall.

A big drawback of these measures is that they are not symmetric. A symmetric alternative is the *relative non-intersecting area (RNIA)*[3] of the two clusterings:

$$\mathrm{RNIA}(\mathcal{S}, \mathcal{S}') = \frac{|U| - |I|}{|U|}. \tag{4.2}$$

For the two clusterings of Fig. 4.1, $\mathrm{RNIA}(\mathcal{S}, \mathcal{S}') = 19/25$.

4.3.5 Comparing Distance Measures

Table 4.1 is a summary of the various properties of CE, Rand index, VI, and RNIA. Rand index and VI have some serious drawbacks: They do not satisfy all metric axioms, and they fail to penalize for the non-intersecting area in certain special cases.[4] RNIA behaves better, even though it does not satisfy all metric axioms either and cannot be used for comparing ordinary clusterings (it always gives zero distance for partitions of the same data set). The properties of CE are superior to the properties of the other distance measures; note specifically that CE is the only metric we have.

We will focus on CE and RNIA in the rest of this thesis, since these two measures have more desirable properties than VI or Rand.

A final point to note is that RNIA simply measures the intersecting area of the two clusterings and loses a lot of information in doing that. In fact, since CE requires one-to-one matching between clusters, but RNIA rewards for all overlaps, we have the following proposition.

Proposition 1. *For all $\mathcal{S}$ and $\mathcal{S}'$ we have $CE(\mathcal{S}, \mathcal{S}') \geq RNIA(\mathcal{S}, \mathcal{S}')$.*

4.4 Comparing Non-Disjoint Clusterings

In a non-disjoint subspace clustering $\mathcal{S}$, some of the clusters share matrix elements. Comparing this kind of clusterings can be handled by duplicating certain matrix elements in a way such that the result clusterings become disjoint. The previously introduced methods can be then applied.

More specifically, consider a matrix element x_{ij} that belongs to the support of $n_{ij}^{\mathcal{S}}$ clusters of the clustering $\mathcal{S}$ and to the support of $n_{ij}^{\mathcal{S}'}$ clusters of the clustering $\mathcal{S}'$. To make the clustering $\mathcal{S}$ disjoint, we need to have $n_{ij}^{\mathcal{S}}$ copies of the matrix element x_{ij}. To make the clustering $\mathcal{S}'$ disjoint, we need to have $n_{ij}^{\mathcal{S}'}$

[3] Also known as the *symmetric difference* of two sets.

[4] Consider clusterings $\mathcal{S}$ and $\mathcal{S}'$. Add a few clusters of size 2 to $\mathcal{S}$, such that these new clusters are not included in $\mathrm{supp}(\mathcal{S}')$. After the addition, $d(\mathcal{S}, \mathcal{S}')$ might decrease, contradicting intuition.

	CE	RNIA	VI	1-Rand
Positive	✓	–	✓	✓
Symmetric	✓	✓	✓	✓
Triangle inequality	✓	✓	–	–
Label permutation invariant	✓	✓	✓	✓
Penalty for non-intersecting area	✓	✓	–	–
Background independent	✓	✓	✓	✓
Scale invariant	✓	✓	–	–
Copy invariant	✓	✓	✓	–
Lower bound	0	0	0	0
Upper bound	1	1	$\log(U)$	1
Requires partitioning	–	–	✓	✓
Multiple cluster coverage penalty	$\frac{K-1}{K}U$	0	$\log(K)$	$\frac{U(K-1)}{K(U-1)}$
Generalizable	✓	✓	✓	–
Handles ordinary clusterings	✓	–	✓	✓
Easy to compute	✓	✓	✓	✓
Intuitive and understandable	✓	✓	✓	✓

Table 4.1: Subspace clustering comparison properties of the Clustering Error (CE), the Relative Non-Intersecting Area (RNIA), the Variation of Information (VI), and 1-Rand. The proofs of the bold-face properties can be found in the Appendix; the rest of the proofs are straightforward.

copies of x_{ij}. To make both clusterings disjoint simultaneously, $\max(n_{ij}^{\mathcal{S}}, n_{ij}^{\mathcal{S}'})$ copies of x_{ij} are needed.

Essentially, the element duplication procedure corresponds to redefining the union size of two clusterings $\mathcal{S}$ and $\mathcal{S}'$ as

$$|U| = \sum_{i,j} \max(n_{ij}^{\mathcal{S}}, n_{ij}^{\mathcal{S}'}) \tag{4.3}$$

and the intersection size as

$$|I| = \sum_{i,j} \min(n_{ij}^{\mathcal{S}}, n_{ij}^{\mathcal{S}'}). \tag{4.4}$$

Plugging in these definitions for $|U|$ and $|I|$, RNIA can be computed straightforwardly using Eq. 4.2. As for CE, the cluster intersection matrix can be formed and its diagonal sum maximized as usual. After this, Eq. 4.1 can be used together with the above definition of $|U|$.

Chapter 5

Comparing Related Types of Clusterings

In this chapter, we first extend our analysis to comparing non-axis-aligned subspace clusterings. The comparison scheme we propose includes axis-aligned subspace clusterings as a special case. Further, we observe that similar principles can be applied to comparing attribute weighted clusterings. Lastly, we address the problem of comparing co-clusterings.

5.1 Comparing Non-Axis-Aligned Subspace Clusterings

Recall that a *non-axis-aligned subspace cluster* S is a pair (R, W), where $R \subseteq \{r_1, r_2, \ldots, r_m\}$ is a subset of the data points and W is a collection of vectors $\{w_1, w_2, \ldots, w_D\}$, $w_i \in \mathbb{R}^p$. The vectors in W form a basis for a subspace of the original p-dimensional data space. We use W also to denote this subspace. A *non-axis-aligned subspace clustering* $\mathcal{S}$ is a collection $\{S_1, S_2, \ldots, S_K\}$ of K non-axis aligned subspace clusters.

To simplify the analysis, we require analogously to the axis-aligned case that the clusters of a non-axis-aligned subspace clustering are disjoint. By this we mean that if two clusters share data points, the associated subspaces must be orthogonal. An extension to the case of non-disjoint clusterings seems possible but complicated, and we leave it for future studies.

We can compare non-axis-aligned subspace clusterings using CE or RNIA just as we compared axis-aligned ones if we first define the size of a cluster and the union and the intersection of two clusters. These are introduced next, by means of the principal angles between two subspaces.

5.1.1 Principal Angles

Let us consider two subspaces of $\mathbb{R}^n$, $\mathcal{F}$ and $\mathcal{G}$, such that $p = \dim \mathcal{F} \geq \dim \mathcal{G} = q \geq 1$. The q *principal angles* $\theta_1, \theta_2, \ldots, \theta_q \in [0, \pi/2]$ can be used to measure the similarity of the subspaces. The angles can be defined sequentially for $k = 1, 2, \ldots, q$ by

$$\cos(\theta_k) = \max_{w \in \mathcal{F}} \max_{v \in \mathcal{G}} w^T v \tag{5.1}$$

with

$$w_k = \arg \max_{w \in \mathcal{F}} \max_{v \in \mathcal{G}} w^T v, \tag{5.2}$$

$$v_k = \arg \max_{v \in \mathcal{G}} \max_{w \in \mathcal{F}} w^T v. \tag{5.3}$$

subject to

$$\begin{aligned} ||w|| = ||v|| = 1, \quad w^T w_i = 0, \quad v^T v_i = 0, \\ i = 1, 2, \ldots, k-1. \end{aligned} \tag{5.4}$$

The vectors $w_1, w_2, \ldots, w_q$ and $v_1, v_2, \ldots, v_q$ are referred to as the *principal vectors*. A pair of vectors (w_i, v_i) is referred to as a *principal pair*.

Let us clarify the definition a bit. Given two subspaces $\mathcal{F}$ and $\mathcal{G}$, we first find vectors $w_1 \in \mathcal{F}$ and $v_1 \in \mathcal{G}$ such that the angle between these vectors is as small as possible. This angle is referred to as the first principal angle θ_1. We now proceed to finding vectors $w_2 \in \mathcal{F}$ and $v_2 \in \mathcal{G}$ such that the angle between these vectors is minimized, given the additional restriction that w_2 has to be orthogonal with w_1, and that v_2 has to be orthogonal with v_1. This gives us the second principal angle θ_2. We continue finding principal angles this way. The vectors $w_1, w_2, \ldots$ form an orthogonal set in the subspace $\mathcal{F}$, and the vectors $v_1, v_2, \ldots$ form an orthogonal set in the subspace $\mathcal{G}$. Due to this restriction, the maximum number of these vectors (and hence principal angles) is naturally $\min(\dim \mathcal{F}, \dim \mathcal{G})$. An illustrative example of principal angles is given in Fig. 5.1.

Singular value decomposition (SVD) is a convenient way to compute the principal angles. Let the matrices $Q_\mathcal{F} \in \mathbb{R}^{n \times p}$ and $Q_\mathcal{G} \in \mathbb{R}^{n \times q}$ contain orthonormal bases for the subspaces $\mathcal{F}$ and $\mathcal{G}$, respectively. The SVD of $Q_\mathcal{F}^T Q_\mathcal{G}$ is

$$Y^T Q_\mathcal{F}^T Q_\mathcal{G} Z = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_q), \tag{5.5}$$

where $Y \in \mathbb{R}^{p \times p}$ and $Z \in \mathbb{R}^{q \times q}$ are orthogonal matrices. The ith singular value σ_i is just the cosine of the ith principal angle, i.e. $\sigma_i = \cos(\theta_i)$.

The cosines of the principal angles are also known as canonical correlations and have important applications for instance in statistics, econometrics, and geology. Principal angles can also be used to solve certain constrained optimization problems. [6, 12, 21, 45]

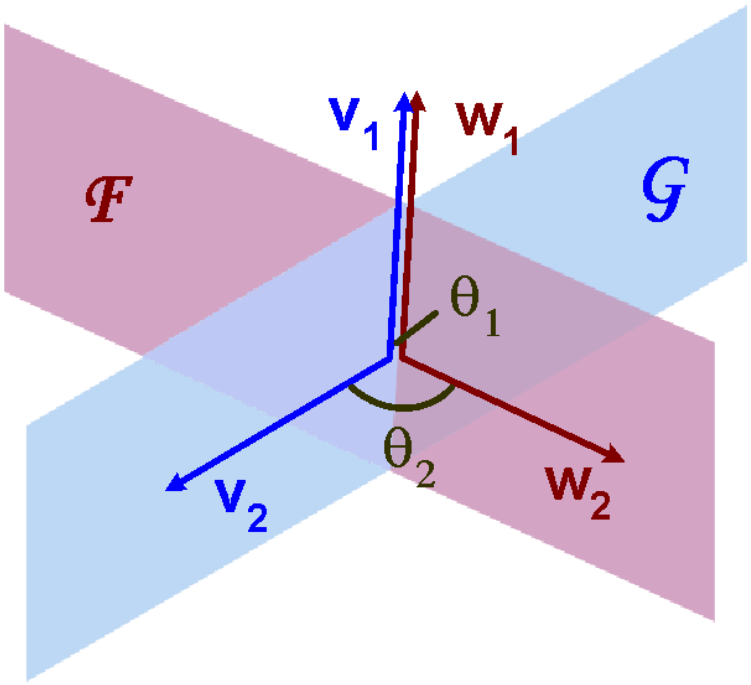


Figure 5.1: Example of principal angles between two subspaces. We have two 2-dimensional subspaces $\mathcal{F}$ and $\mathcal{G}$ in a 3-dimensional space. We thus have two principal angles θ_1 and θ_2 between these subspaces. The first principal angle θ_1 is the angle between the vectors w_1 and v_1; it is naturally zero. The second principal angle θ_2 is the angle between the vectors w_2 and v_2. The vectors w_1, w_2 are orthogonal and lie on the subspace $\mathcal{F}$. Similarly, the vectors v_1, v_2 are orthogonal and lie on the subspace $\mathcal{G}$.

5.1.2 Size, Union, and Intersection of Non-Axis-Aligned Subspace Clusters

In order to use CE, RNIA, or other distance measures with non-axis-aligned subspace clusterings, we need to define the size of a non-axis-aligned subspace cluster and the union and intersection of two such clusters. In case of axis-aligned subspace clusters, our base elements were the matrix elements, or the pairs (data point, attribute). Analogously, the base element here is (data point l, basis vector w).

We count the number of these base elements in a cluster $S_i = (R_i, W_i)$, whose size becomes naturally $m_i = |R_i| \cdot \dim(W_i)$. Continuing with the analogy, we define the intersection of two clusters $S_i = (R_i, W_i) \in \mathcal{S}$ and $S'_j = (R'_j, V_j) \in \mathcal{S}'$ as $m_{ij} = |R_i \cap R'_j| \sum_{k=1}^{q} \sigma_k^2$, where q is the minimum of the dimensions of W_i and V_j and $\{\sigma_k\}$ are the principal angles between W_i and V_j.

This seemingly arbitrary definition for the intersection of two non-axis-aligned subspace clusters has a twofold motivation. First, it is geometrically consistent, as Theorem 1 shows, and in the axis-aligned case it coincides with the previously defined m_{ij} (the number of matrix elements shared by the two clusters). The second, probabilistic, motivation comes from viewing m_{ij}/m_i as the probability of seeing label j in $\mathcal{S}'$ if we randomly pick a point from cluster $S_i \in \mathcal{S}$. This view is used in the original definition of the VI distance and is implicit also in the other distance measures we have considered [40].

Let us investigate the probabilistic motivation in more detail. For any vector w denote by $\Pi_V w$ the projection of w on subspace V. Note that $||\Pi_V w||_2^2 = \cos^2(w, V)||w||_2^2$. We imagine the following sampling process:

1. Pick uniformly a point $l \in R_i$, and if $l \notin R'_j$, stop with 0 successes.
2. Else, pick a random orthonormal basis $\mathcal{B}_{W_i}$ in W_i.
3. For each $w \in \mathcal{B}_{W_i}$, "map w probabilistically to V_j" by counting a success with probability $\cos^2(w, V_j)$ and a failure with probability $\sin^2(w, V_j)$.

It can be shown that the expected number of successes in m_i trials is equal to m_{ij} as defined above.

We have to make sure that the size of the intersection of a subspace cluster S_i with the clusters of the other clustering does not exceed the size of S_i. The following theorem shows this; the proof is presented in the Appendix.

Theorem 1. *Assume that we have a p-dimensional subspace $\mathcal{F}$ and k orthogonal subspaces $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_k$ of dimensionalities $q_1, q_2, \ldots, q_k$. We compute the principal angles $\theta^1_{\mathcal{F},\mathcal{G}_i}, \ldots, \theta^{a_i}_{\mathcal{F},\mathcal{G}_i}$ between all subspace pairs $\mathcal{F}$, $\mathcal{G}_i$; here $a_i = \min(p, q_i)$. It always holds that $\sum_{i=1}^{k} \sum_{j=1}^{a_i} \cos^2(\theta^j_{\mathcal{F},\mathcal{G}_i}) \leq p$. Equality is at-*

tained if and only if $\mathcal{F}$ admits an orthogonal decomposition[1] *over the subspaces* $\{\mathcal{G}\}$.

We have defined the cluster size m_i for a non-axis-aligned subspace cluster. We have also defined the size m_{ij} of the intersection of two non-axis-aligned subspace clusters, motivated this definition, and made sure that it is geometrically consistent. We now proceed to defining the size of the intersection and union of two non-axis-aligned subspace clusterings.

The size of the intersection $|I|$ of two non-axis-aligned subspace clusterings is naturally defined as the sum of the intersections of the cluster pairs: $|I| = \sum_{ij} m_{ij}$. The union size $|U|$ of two clusterings can be defined as $|U| = \sum_i m_i + \sum_j m_j - \sum_{ij} m_{ij}$.

After these definitions, we can calculate the distances between two non-axis-aligned subspace clusterings with CE or RNIA simply by using the familiar Eqs. 4.1 and 4.2. To compute CE, we only need the cluster intersection matrix T, which can be constructed using the cluster intersection sizes m_{ij} and the union size $|U|$. To compute RNIA, only the union size $|U|$ and the intersection size $|I|$ are needed.

5.1.3 Example of Non-Axis-Aligned Subspace Clusterings

Assume that we have two non-axis-aligned subspace clusterings, $S = \{S_1, S_2\}$ and $S' = \{S'_1, S'_2\}$, defined as follows.

$$\begin{aligned}
S_1 &= (\{1,2,3\}, \{[1,1,0,0]^T/\sqrt{2}\}) \\
S_2 &= (\{5,6,7\}, \{[0,0,1,0]^T, \\
&\quad\ [0,1,0,1]^T/\sqrt{2}, [1,-2,0,2]^T/3\}) \\
S'_1 &= (\{2,3,4,5\}, \{[2,0,0,1]^T/\sqrt{(}5), [0,1,0,0]^T\}) \\
S'_2 &= (\{5,6,7\}, \{[0,0,1,0]^T, [-1,0,0,2]^T/\sqrt{5}\})
\end{aligned}$$

We immediately observe that the cluster sizes are $m_1 = 3 \cdot 1 = 3$, $m_2 = 3 \cdot 3 = 9$, $m'_1 = 4 \cdot 2 = 8$, and $m'_2 = 3 \cdot 2 = 6$.

We write the orthonormal basis vectors in the matrices Q_{S_1}, Q_{S_2}, $Q_{S'_1}$, and $Q_{S'_2}$. We need these matrices for calculating the principal angles between the subspaces. For instance, the principal angle $\theta^1_{S_1,S'_1}$ is the only non-zero singular value of the matrix $Q^T_{S_1} Q_{S'_1}$. The principal angles are $\theta^1_{S_1,S'_1} = 0.32$ (18.44°), $\theta^1_{S_1,S'_2} = 1.25$ (71.57°), $\theta^1_{S_2,S'_1} = 0$ (0°), $\theta^2_{S_2,S'_1} = 0.89$ (50.77°), $\theta^1_{S_2,S'_2} = 0$ (0°), and $\theta^2_{S_2,S'_2} = 0.69$ (39.23°).

[1] In other words, the subspace $\mathcal{F}$ is spanned by the collection of the basis vectors for subspaces $\{\mathcal{G}_i\}_i$.

These numbers allow us to compute the intersections between the cluster pairs. For instance, for S_1 and S'_1 we get the intersection size $|I|_{S_1,S'_1} = 2 \cdot \cos(0.32) = 1.90$. Similarly, $|I|_{S_1,S'_2} = 0$, $|I|_{S_2,S'_1} = 1.63$, and $|I|_{S_2,S'_2} = 5.31$. It follows that the union area of the two clusterings is $|U| = \sum_i m_i + \sum_j m'_j - \sum_{i,j} |I|_{S_i,S'_j} = 26 - 8.84 = 17.16$.

Finally, $\text{RNIA}(\mathcal{S}, \mathcal{S}') = (17.16 - 8.84)/17.16 = 0.48$ and $\text{CE}(\mathcal{S}, \mathcal{S}') = (17.16 - 7.21)/17.16 = 0.58$.

5.2 Comparing Attribute Weighted Clusterings

Recall that an *attribute weighted cluster* S is a pair (R, b), where $R \subseteq \{r_1, r_2, \ldots, r_m\}$ is a subset of the data points and b is a vector $[b_1, b_2, \ldots, b_p]^T$, where $b_i \geq 0$ and $\sum_{i=1}^{p} b_i = 1$. The vector b defines an importance weight for each column (attribute). An *attribute weighted clustering* $\mathcal{S}$ is a collection of K attribute weighted clusters $\{S_1, S_2, \ldots, S_K\}$.

We can easily utilize CE and RNIA to compare attribute weighted clusterings if we first define the sizes, the unions and the intersections for this kind of clusterings. We define the size of the cluster (R, b) as $|R|$, the number of data points in the cluster. To define the intersection of two clusters, we first need the intersection of two attribute weight vectors. We define this as

$$b_1 \cap b_2 = 1 - \frac{1}{2} \sum_{i=1}^{p} |b_{1i} - b_{2i}|. \tag{5.6}$$

Note that this could be defined in various ways; The present definition corresponds to the *variation distance* commonly used with discrete probability distributions [5].

It always holds that $0 \leq b_1 \cap b_2 \leq 1$. Now the intersection of two clusters (R_1, b_1) and (R_2, b_2) is given as $|R_1 \cap R_2| \cdot |b_1 \cap b_2|$. To bound the sum of the intersection sizes, we require that if two clusters of a clustering share data points, the inner product of the associated attribute weight vectors has to be zero. We previously introduced an analogous definition for non-axis-aligned subspace clusters.

5.3 Comparing Co-Clusterings

Recall that a *co-clustering* $\mathcal{S} = (\mathcal{R}, \mathcal{C})$ is a simultaneous partitioning of the rows and the columns of the data matrix; $\mathcal{R} = \{R_1, R_2, \ldots, R_L\}$ denotes the

collection of row clusters and $\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$ the collection of column clusters.

Since co-clusterings are always partitionings of the data matrix elements, we can straightforwardly use any of the ordinary clustering distance measures of Section 3.1. For instance, we simply write $\mathrm{VI}(\mathcal{S}, \mathcal{S}')$ for the VI distance between two co-clusterings $\mathcal{S}$ and $\mathcal{S}'$. In calculating the VI (or any other distance) for co-clusterings, we consider each data matrix element as a data point with a cluster label, and the co-clustering as a partition of the data matrix elements.

It is possible to derive relationships for the distances between two co-clusterings and their corresponding row and column clusterings. For instance, the following propositions hold. The proofs are presented in the Appendix.

Proposition 2. *For all co-clusterings $\mathcal{S}, \mathcal{S}'$ we have $VI(\mathcal{S}, \mathcal{S}') = VI(\mathcal{R}, \mathcal{R}') + VI(\mathcal{C}, \mathcal{C}')$.*

Proposition 3. *For all co-clusterings $\mathcal{S}, \mathcal{S}'$ we have $CE(\mathcal{S}, \mathcal{S}') = CE(\mathcal{R}, \mathcal{R}') + CE(\mathcal{C}, \mathcal{C}') - CE(\mathcal{R}, \mathcal{R}')\,CE(\mathcal{C}, \mathcal{C}')$.*

Proposition 4. *For all co-clusterings $\mathcal{S}, \mathcal{S}'$ we have $RNIA(\mathcal{S}, \mathcal{S}') = 0$.*

5.3.1 Example of Co-Clusterings

Consider two co-clusterings $\mathcal{S}$ and $\mathcal{S}'$. We have $R = \{\{1,2,3,4\},\{5,6,7\},\{8\}\}$, $C = \{\{1,2,3\},\{4,5,6\},\{7,8\}$, $R' = \{\{1,2,4\},\{3,6,7\},\{5,8\}\}$, and $C' = \{\{1,2,5,6\},\{3,4,7\},\{8\}\}$. The VI distances are $\mathrm{VI}(\mathcal{R}, \mathcal{R}') = 0.93$, $\mathrm{VI}(\mathcal{C}, \mathcal{C}') = 1.41$, and $\mathrm{VI}(\mathcal{S}, \mathcal{S}') = \mathrm{VI}(\mathcal{R}, \mathcal{R}') + \mathrm{VI}(\mathcal{C}, \mathcal{C}') = 2.34$. The CE distances are $\mathrm{CE}(\mathcal{R}, \mathcal{R}') = 2/8$, $\mathrm{CE}(\mathcal{C}, \mathcal{C}') = 4/8$, and $\mathrm{CE}(\mathcal{S}, \mathcal{S}') = \mathrm{CE}(\mathcal{R}, \mathcal{R}') + \mathrm{CE}(\mathcal{C}, \mathcal{C}') - \mathrm{CE}(\mathcal{R}, \mathcal{R}')\mathrm{CE}(\mathcal{C}, \mathcal{C}') = 40/64$. All RNIA distances are zero.

Chapter 6

Experimental Results

We now demonstrate how our distance measures can be used to calculating distances between clusterings produced by a variety of subspace clustering algorithms on synthetic data sets. In our first experiment, detailed in Section 6.1, we compare the algorithms by the means of external cluster validation. We also show how the use of the subspace clustering distance measures can give more information than traditional row-based or column-based comparison approaches. In Section 6.2, we describe an experiment illustrating how subspace clustering distance measures can be used for studying the stability of various clusterings produced by a single algorithm on a given data set. These results indicate that our distance measures are useful for internal cluster validation.

6.1 External Cluster Validation

6.1.1 Data Sets and Algorithms

We compare the performance of four algorithms, PROCLUS [2], FASTDOC [51], HARP [66], and ORCLUS [3] on synthetic data sets. The first three algorithms produce axis-aligned subspace clusterings, and ORCLUS produces non-axis-aligned clusterings. A description of these algorithms was given in Section 2.3. We compare clusterings produced by these algorithms using our extended CE and RNIA distance measures, which were the two candidates possessing the most desirable theoretical properties (see Section 4.3.5 for details).

We use clustering results from earlier work [65, 66] that compared the algorithms across 8 synthetic data sets. Each data set has 500 data points, 20 attributes, and 5 axis-aligned subspace clusters. The corresponding row clusters form a partition of the data points (rows). The number of data points in each cluster varies from 15% to 25% of the total number of points. The 8 data sets differ in the dimensionality of the subspace clusters. In the first data

set, the dimensionality of all subspace clusters is 4, in the second data set it is 6, and finally in the 8th data set, the subspaces are 18-dimensional. In an attribute relevant to a subspace, the standard deviation of the within-cluster data is between 3% and 5% of the global standard deviation on that attribute.[1] No noise is added. For each data set, we have several clustering results for each algorithm corresponding to various parameter values (except for HARP, which is deterministic and has no input parameters).

6.1.2 Results

Qualitative Comparison of the Algorithms

Some of the clustering results by HARP, PROCLUS, and FASTDOC are visualized in Figs. 6.1 and 6.2, together with the original clusterings.[2] To illustrate the full range of clusterings that we analyzed, we have chosen to plot the best and the worst clusterings produced by each algorithm. Let us first consider the CE distance for subspace clusterings and take a look at the clusterings in Fig. 6.1 (a), Fig. 6.2 (a) and Fig. 6.2 (c). These are the clustering results for the data sets with 4-dimensional, 10-dimensional, and 18-dimensional subspace clusters, respectively.[3]

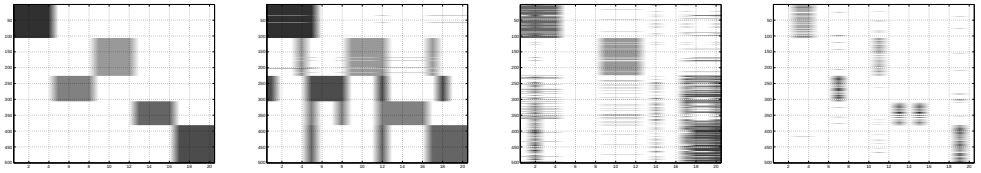
The output clusters by the various algorithms are clearly different from each other. HARP performs reasonably well in all cases. The row clusterings seem right, but in the lower-dimensional data sets, HARP adds extra attributes to the clusters, and in the 18-dimensional data set, some of the relevant attributes are missing. PROCLUS has some difficulty with the 4-dimensional data set, but its clustering for the 10-dimensional data set is almost perfect, and the result for 18-dimensional data set is also good. However, FASTDOC has serious trouble with the column clusters, and a significant number of the data points are not included in the row clusters.

The performance of HARP seems even more impressive when we recall that HARP does not require any input parameters. The clusterings produced by PROCLUS and FASTDOC are strongly dependent on the choice of the input parameters, as Figs. 6.1 (d), 6.2 (b), and 6.2 (d) illustrate. These figures show the worst clusterings produced by PROCLUS and FASTDOC for the three data sets, as judged by the CE distance for subspace clusterings. The worst clusterings by FASTDOC contain tiny clusters with only a few data points and attributes, and the worst clusterings by PROCLUS are not convincing either.

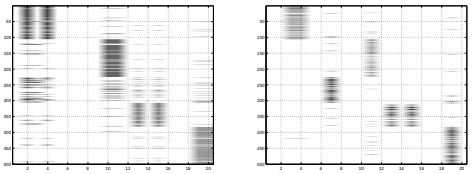
[1] Only the ratio of the standard deviations affects the performance of the algorithms; the magnitudes as such do not have an effect.

[2] Since ORCLUS produces non-axis-aligned clusters, its results cannot be visualized here.

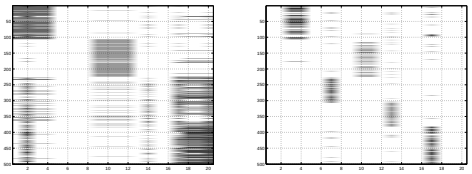
[3] The corresponding results by RNIA are not shown here, since they are very similar to the CE results.



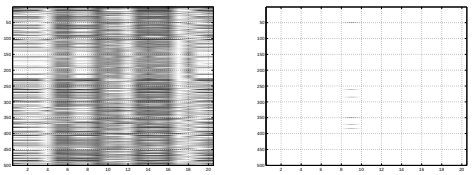
(a) Data set with five 4-dimensional subspace clusters. The original clustering and the best clustering by HARP, PROCLUS, and FASTDOC, decided based on the CE score for subspace clusterings. The CE scores are 0.34, 0.65, and 0.80.



(b) Data set with five 4-dimensional subspace clusters. The best clustering by PROCLUS and FASTDOC, decided based on the CE score for row clusterings. The CE scores are 0.23 and 0.37.

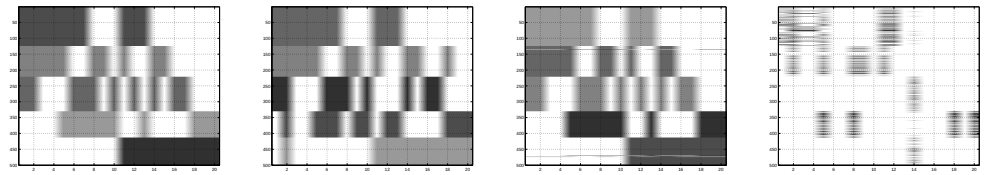


(c) Data set with five 4-dimensional subspace clusters. The best clustering by PROCLUS and FASTDOC, decided based on the CE score for column clusterings. The CE scores are 0.52 and 0.65.

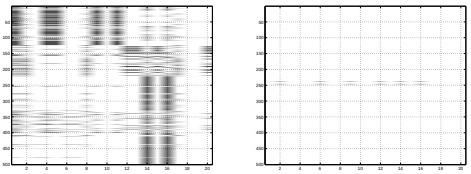


(d) Data set with five 4-dimensional subspace clusters. The worst clustering by PROCLUS and FASTDOC, decided based on the CE score for subspace clusterings. The CE scores are 0.92 and 1.00.

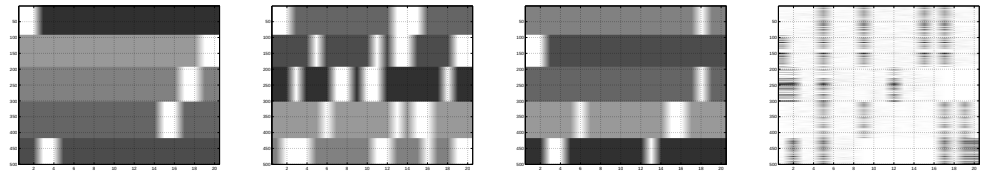
Figure 6.1: Various subspace clusterings for a data set with 500 data points, 20 dimensions, and five 4-dimensional subspace clusters. Each small picture illustrates a subspace clustering. Each subspace cluster is represented by a different shade of gray (the colors do not imply correspondence between clusters of different clusterings), and the unclustered background is white. The definition of the 'best' clustering depends on the choice of the distance for PROCLUS and FASTDOC. We only have one HARP clustering, since HARP is a deterministic algorithm without input parameters.



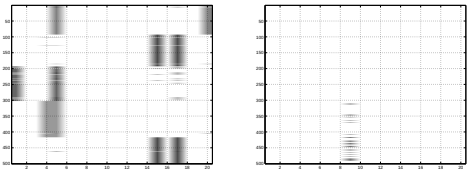
(a) Data set with five 10-dimensional subspace clusters. The original clustering and the best clustering by HARP, PROCLUS, and FASTDOC, decided based on the CE score on subspace clusterings. The CE scores are 0.10, 0.01, and 0.85.



(b) Data set with five 10-dimensional subspace clusters. The worst clustering by PROCLUS and FASTDOC, decided based on the CE score on subspace clusterings. The CE scores are 0.88 and 1.00.



(c) Data set with five 18-dimensional subspace clusters. The original clustering and the best clustering by HARP, PROCLUS, and FASTDOC, decided based on the CE score on subspace clusterings. The CE scores are 0.23, 0.10, and 0.88.



(d) Data set with five 18-dimensional subspace clusters. The worst clustering by PROCLUS and FASTDOC, decided based on the CE score on subspace clusterings. The CE scores are 0.91 and 1.00.

Figure 6.2: Rows 1 and 2: subspace clusterings for a data set with 500 data points, 20 attributes, and five 10-dimensional subspace clusters. Rows 3 and 4: subspace clusterings for a data set with five 18-dimensional subspace clusters. The clusterings produced by HARP, PROCLUS, and FASTDOC are different from each other, especially on the choice of subspaces. The best and the worst clusterings differ significantly; the dependence on the parameter values is strong.

It is therefore of utmost importance to choose the parameter values for these algorithms with care.

Comparing Subspace Clusterings, Row Clusterings, and Column Clusterings

Let us now consider the difference between the CE distance for subspace clusterings, row clusterings, and column clusterings, and take another look at Fig. 6.1. Recall that Fig. 6.1 (a) shows the best clusterings by PROCLUS and FASTDOC according to the CE distance for subspace clusterings. Fig. 6.1 (b) shows the best clusterings according to the CE distance for row clusterings, and finally, Fig. 6.1 (c) shows the best clusterings according to the CE distance for column clusterings. For PROCLUS, the best clusterings given by subspace CE and column CE are equal, but the best clustering given by row CE is different. On the other hand, for FASTDOC, the best clusterings given by subspace CE and row CE are equal, but the result by column CE is different. Thus, the choice of the clustering distance type seems to matter.

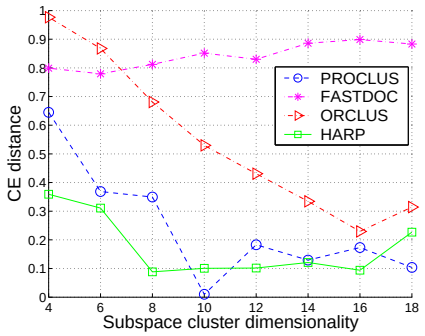
Let us now investigate the question in a quantitative way. Fig. 6.3 shows a comparison between the algorithms on all data sets using six different distance measures.[4] In the first column, we have CE distance for subspace clusterings, CE distance using row information only, and CE distance using column information only.[5] In the second column, the corresponding results for RNIA are shown. Note that the column clusterings are non-disjoint, so we need to apply the element duplication procedure from Section 4.4 here.

The figure clearly brings out differences between the subspace clustering distances, the row clustering distances, and the column clustering distances, indicating that it is indeed worthwhile to pay attention to the choice of the clustering type to compare. For instance, according to the row clustering results, HARP performs well for all data sets and always gives the best result. This is somewhat misleading, since the other two distance measures reveal that HARP's choice of subspaces leaves room for improvement. Thus, irrespective of whether we wish to compare algorithms to each other or analyze the performance of a given algorithm across data sets, the row, column, or subspace based distance measures give different information.

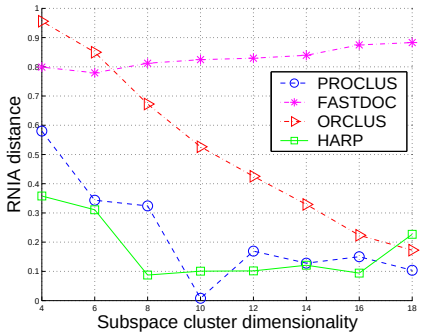
The results of CE and RNIA are very similar on the subspace clustering distances (top row of Fig. 6.3) and on the column clustering distances (bottom

[4]We have chosen to plot only the best clustering results of each algorithm, since the dependence on parameter values is strong, and it does not make sense in this case to plot the means with error bars.

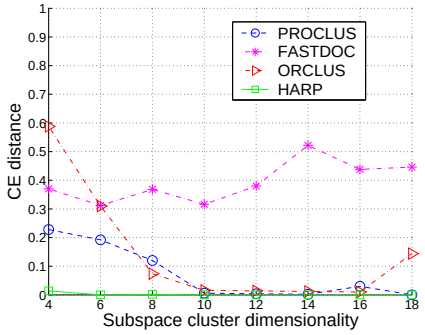
[5]Since we do not yet have a method for handling non-disjoint non-axis-aligned clusterings, the column distance measure for ORCLUS is not shown.



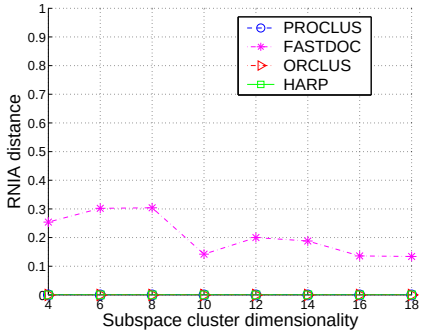
(a) CE distances for subspace clusterings.



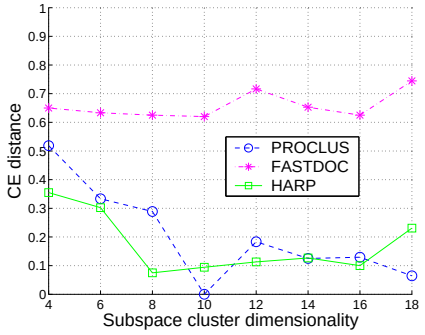
(b) RNIA distances for subspace clusterings.



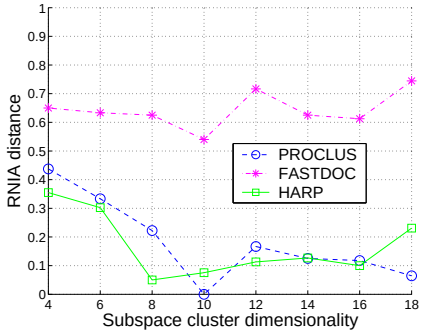
(c) CE distances for row clusterings.



(d) RNIA distances for row clusterings.



(e) CE distances for column clusterings.



(f) RNIA distances for column clusterings.

Figure 6.3: Distances between PROCLUS output and the true clustering, FastDOC output and the true clustering, etc., for eight different data sets (cluster dimensionalities $4, 6, \ldots, 18$). Only the best clustering results of each algorithm are shown. Since we do not yet have a method for handling non-disjoint non-axis-aligned clusterings, the column distance measure for ORCLUS is not shown.

row of Fig. 6.3). However, the RNIA scores for PROCLUS, ORCLUS, and HARP are zero when the row clustering distances are computed. This is natural, since all three algorithms give a partition of the full set of data points, and since the original clustering contains a partition of the same set, we do not have any non-intersecting area between the clusterings. On the other hand, FASTDOC does not produce a partition of the set of data points, which is why its RNIA scores are non-zero.

6.2 Internal Cluster Validation

6.2.1 Data Sets and Algorithms

We conducted another experiment on a new data set consisting of 1000 rows, 100 columns, and 5 subspace clusters. The goal here is to compare the performance of PROCLUS and SSPC [67] over samples of this data set with various parameter settings. Our non-disjoint axis-aligned subspace clusters are 10-dimensional and each cluster has approximately 200 rows. The row clusters form a partition of the set of all rows. The standard deviation of the within-cluster data is between 3% and 5% of the global standard deviation. Five samples of this data set were created by removing 10% of the rows and 10% of the columns.[6] These 5 samples were then clustered by PROCLUS and SSPC. PROCLUS was run with 9 different parameter values for each sample, resulting in 45 clusterings, and SSPC was run with 10 different parameter values per sample, resulting in 50 clusterings.

6.2.2 Results

We have computed the pairwise distances for all 96 clusterings (the true clustering, 50 SSPC clusterings, and 45 PROCLUS clusterings) using the subspace clustering CE distance and the subspace clustering RNIA distance. Fig. 6.4 shows the distances between the true clustering and the other 95 clusterings. In each figure, the clusterings on the left-hand side of the vertical line are the SSPC clusterings, and the right-hand side clusterings are the PROCLUS clusterings. Most SSPC clusterings are closer to the true clustering than the PROCLUS clusterings. The clustering distances exhibit a clear cyclical patterns with respect to samples: for both SSPC and PROCLUS, the dependence of the clustering result on the parameter value is much larger than the dependence on the sample.

[6] An extensive series of experiments would be needed to determine a good sample size and to see whether sampling both the rows and the columns is necessary. In this thesis, we present only preliminary experiments on internal cluster validation.

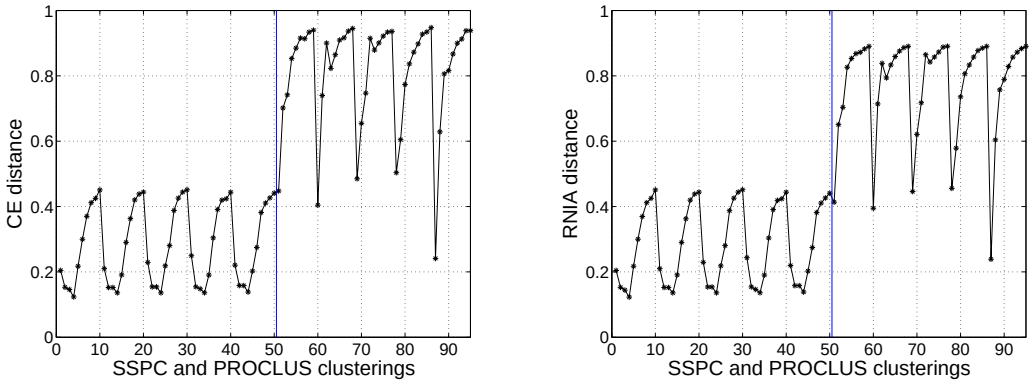
Let us continue investigating the pairwise distances by considering dendrograms, since these might be easier to interpret than visualizations of distance matrices — it is important to note, however, that a lot of information is lost in constructing the dendrograms. Fig. 6.5 (a) shows a single-linkage dendrogram produced by an agglomerative hierarchical clustering algorithm based on the subspace clustering CE distances between all pairs of clusterings. The five groups on the left correspond to the SSPC clusterings on the five samples; the true clustering is included in the fifth group. The PROCLUS clusterings lie on the right-hand side of the dendrogram and do not seem to contain any clear structure.

The subspace CE distance matrix in Fig. 6.5 (e) supports our findings. Clustering 1 is the true clustering, clusterings 2–51 are the SSPC clusterings, and clusterings 52–96 are the PROCLUS clusterings. The SSPC and the PROCLUS clusterings are separated by vertical and horizontal lines. The clusterings are ordered by sample: the clusterings 2–11 are the SSPC clusterings for the first sample, the clusterings 12–21 are the SSPC clusterings for the second sample, and so on. Similarly, the clusterings 52–60 are the PROCLUS clusterings for the first sample, the clusterings 61–69 are the PROCLUS clusterings for the second sample, etc. The graph in Fig. 6.4 (a) corresponds to the first horizontal line of this matrix, excluding the first element. It is clear that the SSPC clusterings are much closer to each other than the PROCLUS clusterings, which do not exhibit clear clustering structure. In each sample, the SSPC clusterings are clustered into two groups by parameter value, as the dendrogram of Fig. 6.5 (a) also shows.

Let us take a look at the same distance matrix ordered by parameter value, as shown in Fig. 6.5 (c). Here, clustering 1 is once again the true clustering. Clusterings 2-6 are the SSPC clusterings for the first parameter value, clusterings 7-11 are the SSPC clusterings for the second parameter value, clusterings 52-56 are the PROCLUS clusterings for the first parameter value, and so on. Now it is clearly visible that the SSPC clusterings form two clusters according to parameter value, and one of these clusters (roughly, the group of clusterings 27–51) is closer to the true clustering. Once again, no structure in the PROCLUS clusterings is visible.

The right column of Fig. 6.5 shows similar results using the subspace RNIA distance. As the dendrogram in Fig. 6.5 (b) illustrates, the RNIA results differ from the CE results. As before, the five clusters on the left-hand side of the dendrogram correspond to the SSPC clusterings for the five samples. However, now also some of the PROCLUS results seem to exhibit clustering structure; five groups of four PROCLUS clusterings each are visible.

The distance matrix in Fig. 6.5 (f) brings more light in the situation. The five groups correspond to the five samples, and in each sample, the clusterings

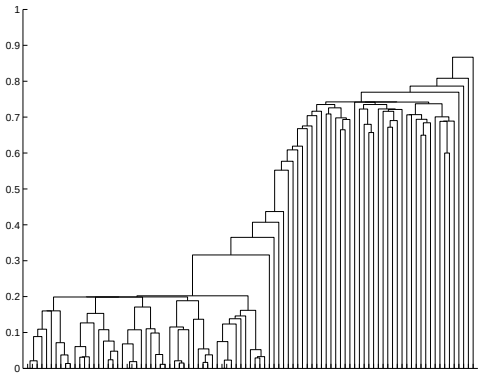


(a) CE distances. (b) RNIA distances.

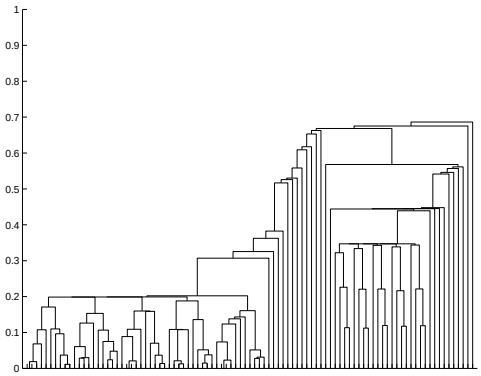
Figure 6.4: The distances between the true clustering and the SSPC and the PROCLUS clusterings. The SSPC clusterings 1–50 lie on the left side of the vertical line; the PROCLUS clusterings 51–95 lie on the right side of the line. The clusterings are ordered by sample: 1–10 are the SSPC clusterings for the first sample, 11–20 are the SSPC clusterings for the second sample, etc. The distances exhibit cyclical patterns w.r.t. the samples.

corresponding to the four last parameter values are clustered together. This is easy to explain. Recall from Section 2.3 that the parameter value required by PROCLUS is the average dimensionality of the output subspace clusters. The parameter values 6-9 correspond to higher dimensionalities 60-90, and if the clusters are so high-dimensional as to fill almost the whole 100-dimensional data matrix, they are bound to be close to each other in the RNIA sense; the relative overlap is very high, no matter how bad the clusters are. However, this undesirable phenomenon does not show in the CE results, since CE requires a one-to-one matching between the clusters. The effect of the PROCLUS parameter value is further noticed in the distance matrix of Fig. 6.5 (f) in that the clusterings corresponding to the smaller parameter values seem to be closest to the true clustering (see the right-hand side of the first row of the matrix). This is to be expected, since the true clusters are 10-dimensional, and the smallest PROCLUS parameter value used was 10.

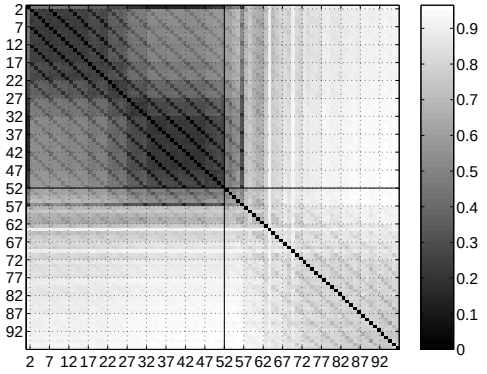
This experiment demonstrates how subspace clustering distance measures could be used for stability-based internal cluster validation for subspace clusterings. We have shown that omputing pairwise distances between clusterings is able to provide us infromation on the stability of algorithms. Based on our experiments, it is clear that SSPC is more stable algorithm than PROCLUS, since the SSPC results vary less across samples. It is important to note that the choice of the subspace clustering distance measure does matter: CE is a



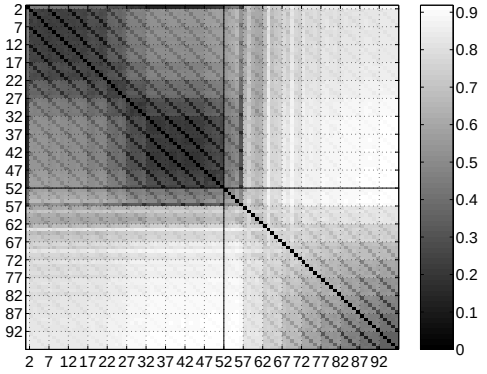
(a) Single-linkage dendrogram, constructed using CE distances for subspace clusterings.



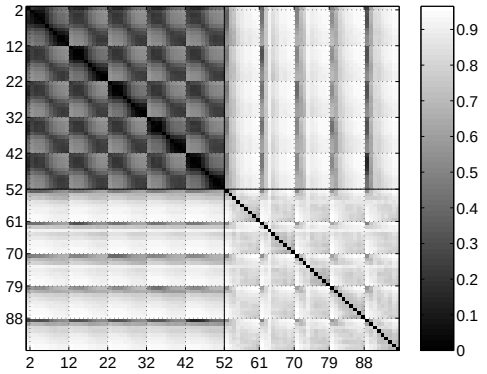
(b) Single-linkage dendrogram, constructed using RNIA distances for subspace clusterings.



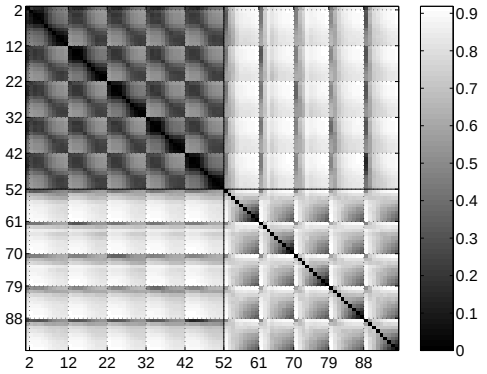
(c) Distance matrix ordered by parameter value, constructed using CE distances for subspace clusterings.



(d) Distance matrix ordered by parameter value, constructed using RNIA distances for subspace clusterings.



(e) Distance matrix ordered by sample, constructed using CE distances for subspace clusterings.



(f) Distance matrix ordered by sample, constructed using RNIA distances for subspace clusterings.

Figure 6.5: Representations of the pairwise distances for 50 clusterings by SSPC (10 parameter values for each of the 5 samples) and 45 clusterings by PROCLUS (9 parameter values for each of the 5 samples). In the distance matrices, 1: True clustering. 2–51: SSPC clusterings. 52–96: PROCLUS clusterings. The horizontal/vertical lines mark the borders between the SSPC clusterings and the PROCLUS clusterings.

better choice than RNIA in the case of high-dimensional subspace clusters.

We have further shown that subspace clustering distance measures are useful in other ways; noticing that the SSPC clusterings corresponding to various parameter values fall into two groups gives us more information on how the SSPC algorithm works; this is an example of meta-clustering.

Chapter 7

Conclusions

In this thesis, we have addressed the problem of comparing subspace clusterings. We have done a comprehensive literature survey on subspace clustering articles and observed that there is currently no satisfactory way to compare subspace clusterings. We have motivated our work by arguing that comparing clusterings is of crucial importance in external and internal cluster validation, meta-clustering, and consensus clusterings; all these topics are reviewed in the thesis.

We have read extensively about the existing methods for comparing ordinary clusterings, presented a summary of them, and justified why these methods cannot be used for comparing subspace clusterings. Since comparing subspace clusterings is more general task than comparing ordinary clusterings, we have introduced a set of theoretical properties important for a subspace clustering distance. We have introduced four candidates for comparing subspace clusterings, namely CE, RNIA, VI, and Rand, and characterized them in terms of their theoretical properties. CE, VI, and Rand are generalizations of existing methods for comparing ordinary clusterings, and RNIA is a novel retrieval measure.

Out of these four distance measure candidates, we have chosen to use CE and RNIA in our experiments, since these two measures possess the most desirable theoretical properties. In the experiments, we have compared clusterings by five well-known algorithms: FASTDOC, HARP, PROCLUS, ORCLUS, and SSPC. We have demonstrated how our measures can be used in both external and internal cluster validation. We have also shown that comparing subspace clusterings gives different information than comparing the corresponding row and column clusterings; we have further argued that comparing the row or the column clusterings is in general not even possible with the ordinary clustering comparison measures. Our experiments have demonstrated that CE is a better choice for a distance measure than RNIA in the case of high-dimensional

subspace clusters.

An additional strength of the CE and RNIA measures is that they can be applied to non-axis-aligned subspace clusterings and attribute weighted clusterings in addition to axis-aligned subspace clusterings. However, the theory of non-axis-aligned subspace clusterings has not been extended to the case of non-disjoint clusterings yet, and we have not shown experiments for comparing attribute weighted clusterings.

As for the other two distance measures, VI, and Rand, the work is not yet finished. There are three main directions to follow. First, we have to recall that the less desirable properties of VI and Rand have only been visible in rare special cases, namely in the cases where the subspace clusterings have several tiny clusters. It might be possible to show that VI and Rand possess more desirable properties if the clusterings satisfy certain conditions, which would hopefully be always satisfied in practice.

Another possible research direction is based on recalling that both VI and Rand require partitioning the non-intersecting area of the clusterings into extra singleton clusters. It might be possible to show that any method that requires the use of extra singleton clusters cannot possibly satisfy all desirable properties, for instance the triangle inequality.

Finally, the third research direction would modify the definitions of VI and Rand so that they would be more suitable for comparing subspace clusterings. For instance, VI could be used on the intersection of the clusterings only, added by an area penalty for the non-intersecting areas. The definition of Rand could be extended to better describe point pairs in the context of subspace clusterings. More specifically, we could classify pairs of points into four classes: the points are in different clusters (class 0); the points are in the same cluster (class 1); one of the points is in a cluster and the other one is in the background (class 2); both points are in the background (class 3). With ordinary clusterings, we only have classes 0 and 1, leading to point pair counts N_{00}, N_{01}, N_{10}, and N_{11} (see Section 3.1.2 for details). With subspace clusterings, we could use all four classes and hence quantities such as N_{22} or N_{03}. A generalized version of the Rand index could be constructed using these quantities. We have not yet fully investigated the theoretical properties of these new distance measures.

It turns out that the distance measures we have proposed for axis-aligned subspace clusterings are useful for comparing other types of clusterings also. In designing the distance measures, we have not restricted ourselves in any way to consider only rectangular sets of matrix elements as clusters. Hence, our distance measures are applicable to any *partial clusterings*: clusterings on subsets of data points. We are not aware of any existing methods for comparing partial clusterings. These kinds of clusterings commonly arise in stability-based internal cluster validation, where we want to compare clusterings on

samples of the data. Most previous approaches have compared the clusterings only at their intersection [10]. Partial clusterings might also arise in distributed databases [54].

Our distance measures could also be useful for comparing hierarchical clusterings. Hierarchical clusterings are commonly described by dendrograms, which in turn can be viewed as non-disjoint clusterings, i.e., clusterings in which a given data point may belong to several clusters. More specifically, a hierarchical clustering for N data points is a collection of N partitions of these N points; each data point thus belongs to N clusters simultaneously. We are currently studying the properties of our distance measures in comparing hierarchical clusterings; one of the issues under investigation is including the dendrogram edge distances in the comparison.

Comparing hierarchical clusterings is interesting, since hierarchies are commonly used in bioinformatics for instance to represent the evolutionary relations of proteins [34]. Also, the subspace clustering algorithms HARP and COSA produce hierarchical clusterings; it would be exciting to see how these clusterings compare to ordinary hierarchical clusterings. Not much work has been done on comparing hierarchical clusterings; current alternatives are a layer-by-layer comparison [25], using the *hop distance* [7], or comparing the so-called *cophenetic matrices* [56].

There are still more avenues to be explored in the future. Weighting the rows and the columns of the data matrix is another potentially useful feature. We have only discussed hard clusterings; probabilistic subspace clusterings would require separate analysis.[1] Finally, depending on the definition of the closeness of the data points in the subspace (for instance, distance-based [2] or pattern-based [63]), the rows and the columns of the data matrix may or may not be symmetric, and a successful comparison method should take this into account.

[1]However, we are only aware of one algorithm producing probabilistic subspace clusterings [49].

Bibliography

[1] P. K. Agarwal and N. H. Mustafa. K-means projective clustering. In *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 155–165, 2004.

[2] C. Aggarwal, C. Procopiuc, J. Wolf, P. Yu, and J. Park. A framework for finding projected clusters in high dimensional spaces. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1999.

[3] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 70–81, 2000.

[4] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 94–105, 1998.

[5] D. Aldous and J. Fill. Reversible Markov Chains and random walks on graphs. `http://www.stat.berkeley.edu/users/aldous/RWG/book.html`.

[6] M. E. Argentati. Principal angles between subspaces. `http://www-math.cudenver.edu/~aknyazev/teaching/rico/talk_defense.pdf`.

[7] P. Artigas, A. Goldenberg, A. Likhodedov, and R. Caruana. Meta clustering. `http://www-2.cs.cmu.edu/~artigas/classproj/mlproj.ps`, 2000.

[8] S. Bandyopadhyay and U. Maulik. Nonparametric genetic clustering: Comparison of validity indices. *IEEE Transactions on Systems, Man, and Cybernetics — Part C: Applications and Reviews*, 31(1), 2001.

[9] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix

approximation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[10] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[11] J. C. Bezdek. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, 28(3), 1998.

[12] A. Björk and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematical Computation*, 27:579–594, 1973.

[13] C. Böhm, K. Kailing, P. Kröger, and A. Zimek. Computing clusters of correlation connected objects. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pages 455–466, 2004.

[14] C. H. Cheng, A. W.-C. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 84–93, 1999.

[15] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, 2000.

[16] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra. Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004.

[17] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[18] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic coclustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[19] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic coclustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[20] C. Domeniconi, D. Papadopoulos, D. Gunopulos, and S. Ma. Subspace clustering of high dimensional data. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004.

[21] Z. Drmac. On principal angles between subspaces of euclidean space. *Siam Journal of Matrix Analysis Applications*, 22(1):173–194, 2000.

[22] R. C. Dubes. How many clusters are best? — an experiment. *Pattern Recognition*, 20(6), 1987.

[23] S. Dudoit and J. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7), 2002.

[24] M. Ester, H-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[25] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[26] J. H. Friedman and J. J. Meulman. Clustering objects on subsets of attributes. *Journal of the Royal Statistical Society B*, 66:1–25, 2004.

[27] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences, USA*, 97:12079–12084, 2000.

[28] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. In *Proceedings of the 21st International Conference on Data Engineering*, 2005.

[29] S. Guha, N. Koudas, A. Marathe, and D. Srivastava. Merging the results of approximate match operations. In *Proceedings of the 30th International Conference on Very Large Data Bases*, pages 636–647, 2004.

[30] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2/3), 2001.

[31] A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[32] A. K. Jain, A. Topchy, M. H. Law, and J. Buhmann. Landscape of clustering algorithms. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 260–263, 2004.

[33] K. Kailing, H.-P. Kriegel, and P. Kröger. Density-connected subspace clustering for high-dimensional data. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 246–257, 2004.

[34] N. Kaplan, O. Sasson, U. Inbar, M. Friedlich, M. Fromer, H. Fleischer, E. Portugaly, N. Linial, and M. Linial. Protonet 4.0: A hierarchical classification of one million protein sequences. *Nucleic Acids Research*, 33:216–218, 2005.

[35] L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data — An Introduction to Cluster Analysis*. John Wiley & Sons, Inc., 1990.

[36] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Research*, 13(4):703–16, April 2003.

[37] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16:1299–1323, 2004.

[38] E. Levine and E. Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 13:2573–2593, 2001.

[39] J. Liu, W. Wang, and J. Yang. A framework for ontology-driven subspace clustering. In *Proceedings of the Tenth ACM SIGKDD International conference on Knowledge Discovery and Data Mining*, 2004.

[40] M. Meila. Comparing clusterings by the variation of information. In *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*, pages 173–187, 2003.

[41] A. A. Melkman and E. Shaham. Sleeved coclustering. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[42] G. W. Milligan, S. C. Soon, and L. M. Sokol. The effect of cluster size, dimensionality, and the number of clusters on recovery of true cluster structure. *IEEE Transactions of Pattern Analysis and Machine Intelligencce*, 5(1):40–47, 1983.

[43] B. Mirkin. *Mathematical classification and clustering*. Kluwer Academic Press, 1996.

[44] H. Nagesh, S. Goil, and A. Choudhary. MAFIA: Efficient and scalable subspace clustering for very large data sets. Technical Report 9906-010, Northwestern University, 1999.

[45] E. Oja and J. Parkkinen. On subspace clustering. In *Prceedings of the Seventh International Conference on Pattern Recognition*, 1984.

[46] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization, Algorithms and Complexity*. Prentice-Hall, 1982.

[47] L. Parsons, E. Haque, and H. Liu. Evaluating subspace clustering algorithms. In *Proceedings of the Fourth SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data and its Applications*, 2004.

[48] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: A review. *ACM SIGKDD Explorations Newsletter*, 6(1), 2004.

[49] A. Patrikainen and H. Mannila. Subspace clustering of high dimensional binary data — a probabilistic approach. In *Proceedings of the Fourth SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data and its Applications*, 2004.

[50] K.S. Pollard and M. J. van der Laan. Statistical inference for simultaneous clustering of gene expression data. *Mathematical Biosciences*, 176(1):99–121, 2002.

[51] C. M. Procopiuc, M. T. Jones, P. K. Agarwal, and T. M. Murali. A Monte carlo algorithm for fast projective clustering. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2002.

[52] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[53] C. Van Rijsbergen. *Information Retrieval*. Butterworths, 2nd edition, 1979.

[54] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partititons. *Journal on Machine Learning Research*, 3:583–617, 2002.

[55] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18:136–144, 2002.

[56] S. Theodoridis and K. Koutroumbas. *Patter Recognition*. Academic Press, 1999.

[57] R. Tibshirani, G. Walther, D. Botstein, and P. Brown. Cluster validation by prediction strength. Technical report, Statistics Department, Stanford University, 2001.

[58] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters via the gap statistic. *Journal of Royal Statistical Society B*, 63(2):411–423, 2001.

[59] A. Topchy, A.K. Jain, and W. Punch. A mixture model for clustering ensembles. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004.

[60] A. Topchy, M. H. Law, A. K. Jain, and A. Fred. Analysis of consensus partition in cluster ensemble. In *Proceedings of The Fourth IEEE International Conference on Data Mining*, 2004.

[61] A. Topchy, B. Minaei-Bidgoli, A.K. Jain, and W. Punch. Adaptive clustering ensembles. In *Proceedings of the 17th International Conference on Pattern Recognition*, pages 272–275, 2004.

[62] D. L. Wallace. Comment. *Journal of the American Statistical Association*, 78(383):569–576, 1983.

[63] J. Yang, W. Wang, H. Wang, and P. S. Yu. Delta-cluster: Capturing subspace correlation in a large data set. In *Proceedings of the 18th IEEE International Conference on Data Engineering*, pages 517–528, 2002.

[64] K. Yeung, D. Haynor, and W. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–316, 2001.

[65] K. Y. Yip. HARP: A practical projected clustering algorithm for mining gene expression data. Master's thesis, The University of Hong Kong, Pokfulam Road, Hong Kong, 2004. `http://www.csis.hku.hk/~ylyip/papers/thesis.pdf`.

[66] K. Y. Yip, D. W. Cheung, and M. K. Ng. HARP: A practical projected clustering algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1387–1397, 2004.

[67] K. Y. Yip, D. W. Cheung, and M. K. Ng. On discovery of extremely low-dimensional clusters using semi-supervised projected clustering. In *Proceedings of the 21st International Conference on Data Engineering*, 2005.

[68] K. Y. Yip, M. K. Ng, and D. W. Cheung. A review on projected clustering algorithms. *International Journal of Applied Mathematics*, 13:35–47, 2003.

Appendix A

Appendix: Proofs

A.1 Appendix 1: Proofs for Table 1

A.1.1 Preliminaries

In this section, we review our notation and derive new formulas for VI and the Rand index. For CE and RNIA, we use the familiar expressions given in Eqs. 4.1 and 4.2, respectively.

Cluster Sizes and Intersections

Consider two subspace clusterings $\mathcal{S} = (S_1, \ldots, S_K)$ and $\mathcal{S}' = (S'_1, \ldots, S'_{K'})$. The total number of matrix elements in the union of the two clusters is $m = |U|$. The cluster sizes are $m_1, \ldots, m_K$ and $m'_1, \ldots, m'_{K'}$. We write m_{ij} for the cluster intersection size $|S_i \cap S'_j|$.

Variation of Information

To compute the VI distance for two subspace clusterings, we have used Eq. 3.12 after filling the non-intersecting area of these clusterings with singleton clusters. We now show how to derive a simplified version of the VI distance for subspace clusterings.

Consider two axis-aligned subspace clusterings $\mathcal{S} = \{S_1, S_2, \ldots, S_K\}$ and $\mathcal{S}' = \{S'_1, S'_2, \ldots, S'_{K'}\}$. We fill the non-intersecting areas of these clusterings with singleton clusters. The confusion matrix of the resulting clusterings is shown in Fig. A.1.

In the confusion matrix, $m_{0i} = m_i - \sum_{j=1}^{K'} m_{ij}$, i.e., the number of matrix elements in cluster S_i which are not covered by any cluster of $\mathcal{S}'$. We have written S_{0i} to denote the collection of the resulting singleton clusters. Similarly, $m'_{0j} = m_j - \sum_{i=1}^{K} m_{ij}$.

	S'_1	S'_2	...	$S'_{K'}$	$\overbrace{S_{01}}^{m_{01}}$	$\overbrace{S_{02}}^{m_{02}}$	...	$\overbrace{S_{0k}}^{m_{0K}}$	
S_1	m_{11}	m_{12}	...	$m_{1K'}$	1	0	...	0	m_1
S_2	m_{21}	m_{22}	...	$m_{2K'}$	0	1	...	0	m_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
S_K	m_{K1}	m_{K2}	...	$m_{KK'}$	0	0	...	1	$m_{K'}$
m'_{01} S'_{01}	1	0	...	0	0	0	...	0	1
m'_{02} S'_{02}	0	1	...	0	0	0	...	0	1
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	1
$m'_{0K'}$ $S'_{0K'}$	0	0	...	1	0	0	0	0	1
	m'_1	m'_2	...	m'_K	1	1	1	1	$\lvert U\rvert$

Figure A.1: Confusion matrix for two subspace clusterings $\mathcal{S} = \{S_1, S_2, \ldots, S_K\}$ and $\mathcal{S}' = \{S'_1, S'_2, \ldots, S'_{K'}\}$ with the non-intersecting area filled with singleton clusters. In order to express the matrix in a compact form, we have written S_{01} for the the collection of m_{01} singleton clusters; properly expanded confusion matrix would have m_{01} columns with '1' as the topmost element and '0' as the other elements. Similarly, S_{02}, S_{03}, S'_{01}, and others could be expanded out as multiple columns and rows.

We calculate the VI score of Eq.3.12 for this confusion matrix as follows.

$$\begin{aligned}
\mathrm{VI}(\mathcal{S},\mathcal{S}') =& \frac{1}{|U|}\left(\sum_{i=1}^{K}\sum_{j=1}^{K'} m_{ij}\log\frac{m_i m'_j}{m_{ij}^2} + \sum_{i=1}^{K} m_{i0}\log m_i + \sum_{j=1}^{K'} m'_{j0}\log m'_j\right)\\
=& \frac{1}{|U|}\left(\sum_{i=1}^{K}\sum_{j=1}^{K'} m_{ij}\log\frac{m_i m'_j}{m_{ij}^2} + \sum_{i=1}^{K} m_i\log m_i + \sum_{j=1}^{K'} m'_j\log m'_j\right.\\
&\left. - \sum_{i=1}^{K}\sum_{j=1}^{K'} m_{ij}\log m_i - \sum_{j=1}^{K'}\sum_{i=1}^{K} m'_{ij}\log m'_i\right).
\end{aligned}$$

Our expression does not contain any terms related to the unit clusters. We simplify it further, obtaining

$$\begin{aligned}
&\mathrm{VI}(\mathcal{S},\mathcal{S}')\\
&= \frac{1}{|U|}\left(\sum_{i=1}^{K} m_i\log m_i + \sum_{j=1}^{K'} m'_j\log m'_j - 2\sum_{i=1}^{K}\sum_{j=1}^{K'} m_{ij}\log m_{ij}\right).
\end{aligned} \tag{A.1}$$

Note that all we need to compute the VI distance are the cluster size vectors $\mathbf{m} = (m_i)$ and $\mathbf{m}' = (m'_i)$ together with the *cluster intersection matrix* $T =$

(m_{ij}). An essential difference between T and the confusion matrix is that the cluster intersection matrix contains the intersections of the original clusters only, whereas the confusion matrix contains also the unit clusters. It is also important to note that $\mathbf{m}$ and $\mathbf{m}'$ are not marginals of the cluster intersection matrix. Only in the special case of fully intersecting clusters we have $m_i = \sum_{j=1}^{K'} m_{ij}$ and $m'_j = \sum_{i=1}^{K} m_{ij}$. In this case, we recover the original VI.

Rand

Using the confusion matrix, the quantity (1-Rand) can be written as

$$\begin{aligned} 1 - \mathrm{Rand}(\mathcal{S}, \mathcal{S}') &= \frac{N_{10} + N_{01}}{N} \\ &= \frac{\sum_{i=1}^{L'} \sum_{j=1}^{L} \sum_{k=j+1}^{L} m_{ij} m_{ik} + \sum_{i=1}^{L} \sum_{j=1}^{L'} \sum_{k=j+1}^{L'} m_{ji} m_{ki}}{m(m-1)/2}. \end{aligned} \tag{A.2}$$

Another way to write (1-Rand) is

$$1 - \mathrm{Rand}(\mathcal{S}, \mathcal{S}') = \frac{\sum_{i=1}^{L} m_i^2 + \sum_{j=1}^{L'} m_j'^2 - 2 \sum_{i=1}^{L} \sum_{j=1}^{L'} m_{ij}^2}{m(m-1)}. \tag{A.3}$$

In these two equations, $L \geq K$ and $L' \geq K'$ stand for the number of clusters in $\mathcal{S}$ and $\mathcal{S}'$ after filling the non-intersecting area with singleton clusters.

A.1.2 Triangle Inequality

Theorem 2. *(Triangle inequality for CE.) CE$(\mathcal{A}, \mathcal{B}) \leq$ CE$(\mathcal{A}, \mathcal{C}) +$ CE$(\mathcal{B}, \mathcal{C})$ for any subspace clusterings $\mathcal{A}, \mathcal{B}, \mathcal{C}$.*

To prove this theorem, we first show some preliminary results. For simplicity, we adopt a shorthand notation and write $\mathcal{A}$ instead of supp $(\mathcal{A})$, $\mathcal{A} \backslash \mathcal{B}$ instead of supp $(\mathcal{A}) \setminus$ supp $(\mathcal{B})$, etc. Also, we write $\mathcal{A}_{\mathcal{A} \cap \mathcal{B}}$ for the part of the clustering $\mathcal{A}$ in $\mathcal{A} \cap \mathcal{B}$, $\mathcal{A}_{\mathcal{A} \backslash \mathcal{B}}$ for the part of the clustering $\mathcal{A}$ in $\mathcal{A} \backslash \mathcal{B}$, etc. Also, let us define a *cluster label vector* u for a clustering of mp points and K clusters as a vector of size $mp \times 1$ where $u_i = k$ if the ith point belongs to the kth cluster. Here $i \in \{1, 2, \ldots, mp\}$ and $k \in \{1, 2, \ldots, K\}$.

Proposition 5. *If $\mathcal{A}$ and $\mathcal{B}$ are arbitrary subspace clusterings and $\mathcal{C}' \subseteq (\mathcal{A} \cup \mathcal{B})$, then CE$(\mathcal{A}, \mathcal{B}) \leq$ CE$(\mathcal{A}, \mathcal{C}') +$ CE$(\mathcal{B}, \mathcal{C}')$.*

Proof. (Proposition 5) Let us write $\mathrm{H}(u, v)$ for the Hamming distance between two cluster label vectors u and v, or in other words, the total number of differences between these two vectors. Then the CE distance becomes

$$\mathrm{CE}(\mathcal{A}, \mathcal{B}) = \frac{\min \mathrm{H}(\mathcal{A}_{\mathcal{A} \cap \mathcal{B}}, \mathcal{B}_{\mathcal{A} \cap \mathcal{B}}) + |\mathcal{A} \backslash \mathcal{B}| + |\mathcal{B} \backslash \mathcal{A}|}{|\mathcal{A} \cup \mathcal{B}|},$$

where the minimum is taken over all permutations of the cluster labels. Let us consider three subspace clusterings $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}'$ such that $\mathcal{A}$ and $\mathcal{B}$ are arbitrary and $\mathcal{C}' \subseteq (\mathcal{A} \cup \mathcal{B})$. We consider $\mathcal{C}'$ in three disjoint parts: $\mathcal{C}' = \mathcal{C}'_{\mathcal{A}\cap\mathcal{B}} \cup \mathcal{C}'_{\mathcal{A}\setminus\mathcal{B}} \cup \mathcal{C}'_{\mathcal{B}\setminus\mathcal{A}}$. Let us fix the cluster labels of $\mathcal{C}'$ and choose the permutation of labels in $\mathcal{A}$ to minimize $\mathrm{H}(\mathcal{A}_{\mathcal{A}\cap\mathcal{C}'}, \mathcal{C}'_{\mathcal{A}\cap\mathcal{C}'})$ and the permutation of labels in $\mathcal{B}$ to minimize $\mathrm{H}(\mathcal{B}_{\mathcal{B}\cap\mathcal{C}'}, \mathcal{C}'_{\mathcal{B}\cap\mathcal{C}'})$. Using these labels, we have

$$\begin{aligned}
&\mathrm{CE}(\mathcal{A},\mathcal{C}') + \mathrm{CE}(\mathcal{B},\mathcal{C}') - \mathrm{CE}(\mathcal{A},\mathcal{B}) \\
&\geq \frac{\mathrm{H}(\mathcal{A}_{C'_{\mathcal{A}\cap\mathcal{B}}}, \mathcal{C}'_{A\cap B}) + \mathrm{H}(\mathcal{A}_{C'_{\mathcal{A}\setminus\mathcal{B}}}, \mathcal{C}'_{\mathcal{A}\setminus\mathcal{B}})}{|\mathcal{A}\cup\mathcal{C}'|} \\
&\quad + \frac{|\mathcal{A}| - |\mathcal{C}'_{\mathcal{A}\cap\mathcal{B}}| - |\mathcal{C}'_{\mathcal{A}\setminus\mathcal{B}}| + |\mathcal{C}'_{\mathcal{B}\setminus\mathcal{A}}|}{|\mathcal{A}\cup\mathcal{C}'|} \\
&\quad + \frac{\mathrm{H}(\mathcal{B}_{C'_{\mathcal{A}\cap\mathcal{B}}}, \mathcal{C}'_{\mathcal{A}\cap\mathcal{B}}) + \mathrm{H}(\mathcal{B}_{C'_{\mathcal{B}\setminus\mathcal{A}}}, \mathcal{C}'_{\mathcal{B}\setminus\mathcal{A}})}{|\mathcal{B}\cup\mathcal{C}'|} \\
&\quad + \frac{|\mathcal{B}| - |\mathcal{C}'_{\mathcal{A}\cap\mathcal{B}}| - |\mathcal{C}'_{\mathcal{B}\setminus\mathcal{A}}| + |\mathcal{C}'_{\mathcal{A}\setminus\mathcal{B}}|}{|\mathcal{B}\cup\mathcal{C}'|} \\
&\quad - \frac{\mathrm{H}(\mathcal{A}_{\mathcal{A}\cap\mathcal{B}}, \mathcal{B}_{\mathcal{A}\cap\mathcal{B}}) + |\mathcal{A}\setminus\mathcal{B}| + |\mathcal{B}\setminus\mathcal{A}|}{|\mathcal{A}\cup\mathcal{B}|}.
\end{aligned}$$

Above, the first two terms correspond to $\mathrm{CE}(\mathcal{A},\mathcal{C}')$ and the second two terms correspond to $\mathrm{CE}(\mathcal{B},\mathcal{C}')$. Due to the choice of the labels, the quantity in the third term is greater than or equal to $\mathrm{CE}(\mathcal{A},\mathcal{B})$, hence the inequality.

Next, we notice that $|\mathcal{A}| = |\mathcal{A}\cap\mathcal{B}| + |\mathcal{A}\setminus\mathcal{B}|$, that $|\mathcal{B}| = |\mathcal{A}\cap\mathcal{B}| + |\mathcal{B}\setminus\mathcal{A}|$, and that $\mathrm{H}(\mathcal{A}_{\mathcal{A}\cap\mathcal{B}}, \mathcal{B}_{\mathcal{A}\cap\mathcal{B}}) \leq |\mathcal{A}\cap\mathcal{B}| + \mathrm{H}(\mathcal{A}_{C'_{\mathcal{A}\cap\mathcal{B}}}, \mathcal{B}_{C'_{\mathcal{A}\cap\mathcal{B}}}) - |\mathcal{C}'_{\mathcal{A}\cap\mathcal{B}}|$. We substitute these in the above equation, rearrange the terms, and obtain

$$\begin{aligned}
&\mathrm{CE}(\mathcal{A},\mathcal{C}') + \mathrm{CE}(\mathcal{B},\mathcal{C}') - \mathrm{CE}(\mathcal{A},\mathcal{B}) \\
&\geq \frac{\mathrm{H}(\mathcal{A}_{C'_{\mathcal{A}\cap\mathcal{B}}}, \mathcal{C}'_{A\cap B})}{|\mathcal{A}\cup\mathcal{C}'|} + \frac{\mathrm{H}(\mathcal{B}_{C'_{\mathcal{A}\cap\mathcal{B}}}, \mathcal{C}'_{A\cap B})}{|\mathcal{B}\cup\mathcal{C}'|} \\
&\quad - \frac{\mathrm{H}(\mathcal{A}_{C'_{\mathcal{A}\cap\mathcal{B}}}, \mathcal{B}_{C'_{A\cap B}})}{|\mathcal{A}\cup\mathcal{B}|} \\
&\quad + \frac{\mathrm{H}(\mathcal{A}_{C'_{\mathcal{A}\setminus\mathcal{B}}}, \mathcal{C}'_{\mathcal{A}\setminus\mathcal{B}})}{|\mathcal{A}\cup\mathcal{C}'|} + \frac{\mathrm{H}(\mathcal{B}_{C'_{\mathcal{B}\setminus\mathcal{A}}}, \mathcal{C}'_{\mathcal{B}\setminus\mathcal{A}})}{|\mathcal{B}\cup\mathcal{C}'|} \\
&\quad + \frac{(|\mathcal{A}\cap\mathcal{B}| - |\mathcal{C}'_{\mathcal{A}\cap\mathcal{B}}|) + (|\mathcal{A}\setminus\mathcal{B}| - |\mathcal{C}'_{\mathcal{A}\setminus\mathcal{B}}|) + |\mathcal{C}'_{\mathcal{B}\setminus\mathcal{A}}|}{|\mathcal{A}\cup\mathcal{C}'|} \\
&\quad + \frac{(|\mathcal{A}\cap\mathcal{B}| - |\mathcal{C}'_{\mathcal{A}\cap\mathcal{B}}|) + (|\mathcal{B}\setminus\mathcal{A}| - |\mathcal{C}'_{\mathcal{B}\setminus\mathcal{A}}|) + |\mathcal{C}'_{\mathcal{A}\setminus\mathcal{B}}|}{|\mathcal{B}\cup\mathcal{C}'|} \\
&\quad - \frac{|\mathcal{A}\setminus\mathcal{B}| + |\mathcal{B}\setminus\mathcal{A}| + (|\mathcal{A}\cap\mathcal{B}| - |C'_{\mathcal{A}\cap\mathcal{B}}|)}{|\mathcal{A}\cup\mathcal{B}|}.
\end{aligned}$$

We know that the triangle inequality holds for CE with ordinary clusterings (partitions), so in the equation above, the first three terms sum up to 0 or more. The Hamming distance is always nonnegative, so the fourth and the fifth terms above are also greater than or equal to 0. We also notice that $|\mathcal{A}\backslash\mathcal{B}|-|\mathcal{C}'_{\mathcal{A}\backslash\mathcal{B}}| \geq 0$ and $|\mathcal{B}\backslash\mathcal{A}| - |\mathcal{C}'_{\mathcal{B}\backslash\mathcal{A}}| \geq 0$. Finally, it holds that $|\mathcal{A}\cap\mathcal{B}| - |C'_{\mathcal{A}\cap\mathcal{B}}| \geq 0$. These observations lead us to

$$\begin{aligned}&\mathrm{CE}(\mathcal{A},\mathcal{C}') + \mathrm{CE}(\mathcal{B},\mathcal{C}') - \mathrm{CE}(\mathcal{A},\mathcal{B})\\ &\quad\geq \frac{(|\mathcal{A}\backslash B| - |\mathcal{C}'_{\mathcal{A}\backslash\mathcal{B}}|) + |\mathcal{C}'_{\mathcal{B}\backslash\mathcal{A}}|}{|\mathcal{A}\cup\mathcal{C}'|} + \frac{(|\mathcal{B}\backslash\mathcal{A}| - |\mathcal{C}'_{\mathcal{B}\backslash\mathcal{A}}|) + |\mathcal{C}'_{\mathcal{A}\backslash\mathcal{B}}|}{|\mathcal{B}\cup\mathcal{C}'|}\\ &\qquad- \frac{|\mathcal{A}\backslash\mathcal{B}| + |\mathcal{B}\backslash\mathcal{A}|}{|\mathcal{A}\cup\mathcal{B}|}.\end{aligned}$$

We lastly observe that $|A\cup\mathcal{B}| \geq |A\cup\mathcal{C}'|$ and that $|A\cup\mathcal{B}| \geq |B\cup\mathcal{C}'|$, which helps to complete the proof:

$$\begin{aligned}&\mathrm{CE}(\mathcal{A},\mathcal{C}') + \mathrm{CE}(\mathcal{B},\mathcal{C}') - \mathrm{CE}(\mathcal{A},\mathcal{B})\\ &\quad\geq \frac{|\mathcal{A}\backslash B| - |\mathcal{C}'_{\mathcal{A}\backslash\mathcal{B}}| + |\mathcal{C}'_{\mathcal{B}\backslash\mathcal{A}}| + |\mathcal{B}\backslash\mathcal{A}| - |\mathcal{C}'_{\mathcal{B}\backslash\mathcal{A}}| + |\mathcal{C}'_{\mathcal{A}\backslash\mathcal{B}}| - |\mathcal{A}\backslash\mathcal{B}| + |\mathcal{B}\backslash\mathcal{A}|}{|\mathcal{A}\cup\mathcal{B}|}\\ &\quad= 0.\end{aligned}$$

□

Proposition 6. *If $\mathcal{A}$ and $\mathcal{B}$ are arbitrary subspace clusterings and $\mathcal{C} = \mathcal{C}'\cup\mathcal{C}''$, where $\mathcal{C}' \subset (\mathcal{A}\cup\mathcal{B})$ and $\mathcal{C}''\cap(\mathcal{A}\cup\mathcal{B}) = \emptyset$, then $CE(\mathcal{A},\mathcal{C}) \geq CE(\mathcal{A},\mathcal{C}')$ and $CE(\mathcal{B},\mathcal{C}) \geq CE(\mathcal{B},\mathcal{C}')$.*

Proof. (Proposition 6)

$$\begin{aligned}\mathrm{CE}(\mathcal{A},\mathcal{C}) &= \frac{\mathrm{H}(\mathcal{A}_{\mathcal{A}\cap\mathcal{C}'},\mathcal{C}'_{\mathcal{A}\cap\mathcal{C}'}) + |\mathcal{A}\backslash\mathcal{C}'| + |\mathcal{C}'\backslash\mathcal{A}| + |\mathcal{C}''|}{|\mathcal{A}\cap\mathcal{C}'| + |\mathcal{A}\backslash\mathcal{C}'| + |\mathcal{C}'\backslash\mathcal{A}| + |\mathcal{C}''|}\\ &\geq \frac{\mathrm{H}(\mathcal{A}_{\mathcal{A}\cap\mathcal{C}'},\mathcal{C}'_{\mathcal{A}\cap\mathcal{C}'}) + |\mathcal{A}\backslash\mathcal{C}'| + |\mathcal{C}'\backslash\mathcal{A}|}{|\mathcal{A}\cap\mathcal{C}'| + |\mathcal{A}\backslash\mathcal{C}'| + |\mathcal{C}'\backslash\mathcal{A}|}\\ &= \mathrm{CE}(\mathcal{A},\mathcal{C}').\end{aligned}$$

The case of $\mathcal{B}$ and $\mathcal{C}$ can be proven analogously. □

Proof. (Theorem 2) Let us choose arbitrary subspace clusterings $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, for which $\mathcal{C} = \mathcal{C}'\cup\mathcal{C}''$ such that $\mathcal{C}' \subseteq (\mathcal{A}\cup\mathcal{B})$ and $\mathcal{C}''\cap(\mathcal{A}\cup\mathcal{B}) = \emptyset$. By Propositions 5 and 6, $\mathrm{CE}(\mathcal{A},\mathcal{C}) + \mathrm{CE}(\mathcal{B},\mathcal{C}) \geq \mathrm{CE}(\mathcal{A},\mathcal{C}') + \mathrm{CE}(\mathcal{B},\mathcal{C}') \geq \mathrm{CE}(\mathcal{A},\mathcal{B})$. □

Theorem 3. *(Triangle inequality for RNIA.)* $RNIA(\mathcal{A},\mathcal{B}) \leq RNIA(\mathcal{A},\mathcal{C}) + RNIA(\mathcal{B},\mathcal{C})$ *for any subspace clusterings* $\mathcal{A},\mathcal{B},\mathcal{C}$.

Proof. (Theorem 3) Let us pick subspace clusterings $\mathcal{A},\mathcal{B},\mathcal{C}$ for a data matrix $X = (x_{ij})$. Let us write n_A for the number of the elements of X that are clustered only by $\mathcal{A}$, n_{AB} for the number of the elements of X that are clustered by $\mathcal{A}$ and $\mathcal{B}$ but not $\mathcal{C}$, and n_{ABC} for the number of the elements of X that are clustered by all three clusterings. We define n_B, n_C, n_{AC}, and n_{BC} similarly. Now, we can write

$$\begin{aligned}
&\mathrm{RNIA}(\mathcal{A},\mathcal{C}) + \mathrm{RNIA}(\mathcal{B},\mathcal{C}) - \mathrm{RNIA}(\mathcal{A},\mathcal{B}) \\
&= \frac{n_A + n_C + n_{AB} + n_{BC}}{n_A + n_C + n_{AB} + n_{BC} + n_{AC} + n_{ABC}} \\
&+ \frac{n_B + n_C + n_{AC} + n_{AB}}{n_B + n_C + n_{AC} + n_{AB} + n_{BC} + n_{ABC}} \\
&- \frac{n_A + n_B + n_{AC} + n_{BC}}{n_A + n_B + n_{AC} + n_{BC} + n_{AB} + n_{ABC}}.
\end{aligned}$$

Once the expression is fully expanded, all negative terms disappear. Since the expression is always greater than zero, the triangle inequality holds. □

Example 1. *(Triangle inequality for VI.) VI does not satisfy the triangle inequality in the case of subspace clusterings.*

We show by counterexample that VI does not satisfy the triangle inequality. Consider three subspace clusterings $\mathcal{A} = (A_1) = (\{1,2\},\{1,2,3,4\})$, $\mathcal{B} = (B_1) = (\{2,3\},\{1,2,3,4\})$, and $\mathcal{C} = (C_1, C_2) = (\{1,2\},\{1,2,3,4\},\{1,2\},\{6\})$. We have $\mathrm{VI}(\mathcal{A},\mathcal{B}) = 8/3\log 2 \approx 1.84$, $\mathrm{VI}(\mathcal{A},\mathcal{C}) = 1/5\log 2 \approx 0.14$, and $\mathrm{VI}(\mathcal{B},\mathcal{C}) = 17/7\log 2 \approx 1.68$. Thus $\mathrm{VI}(\mathcal{A},\mathcal{C}) + \mathrm{VI}(\mathcal{B},\mathcal{C}) < \mathrm{VI}(\mathcal{A},\mathcal{B})$, showing that the triangle inequality does not hold. It does not hold for these three clusterings even for VI distances scaled by the logarithm of the union area.

Example 2. *(Triangle inequality for Rand.) The Rand index does not satisfy the triangle inequality in the case of subspace clusterings.*

We show by counterexample that (1-Rand) does not satisfy the triangle inequality. Consider the example clusterings from Example 1. For these clusterings, we have $1 - \mathrm{Rand}(\mathcal{A},\mathcal{B}) = (22+22)/66 \approx 0.67$, $1 - \mathrm{Rand}(\mathcal{A},\mathcal{C}) = 1/45 \approx 0.02$, and $1 - \mathrm{Rand}(\mathcal{B},\mathcal{C}) = 45/91 \approx 0.49$. The triangle inequality does not hold, since $1 - \mathrm{Rand}(\mathcal{A},\mathcal{C}) + 1 - \mathrm{Rand}(\mathcal{B},\mathcal{C}) < 1 - \mathrm{Rand}(\mathcal{A},\mathcal{B})$.

A.1.3 Penalty for Non-Intersecting Area

Consider adding $k \geq 1$ units of non-intersecting area to two subspace clusterings $\mathcal{A}$ and $\mathcal{B}$ and denote the resulting clusterings by $\mathcal{A}^U$ and $\mathcal{B}^U$ (note that

one of these might actually equal to the original clustering). Are our distance measures able to penalize for this added non-intersecting area?

Theorem 4. *(Penalty for Non-Intersecting Area with CE.) CE*$(\mathcal{A}^U, \mathcal{B}^U) \geq$ *CE*$(\mathcal{A}, \mathcal{B})$ *for all subspace clusterings* $\mathcal{A}$*,* $\mathcal{B}$*.*

Proof.

$$\begin{aligned}\mathrm{CE}(\mathcal{A}^U, \mathcal{B}^U) &= \frac{(|U| + k) - D_{max}}{(|U| + k)} \\ &\geq \frac{|U| - D_{max}}{|U|} \\ &= \mathrm{CE}(\mathcal{A}, \mathcal{B}).\end{aligned}$$

□

Theorem 5. *(Penalty for Non-Intersecting Area with RNIA.) RNIA*$(\mathcal{A}^U, \mathcal{B}^U) \geq$ *RNIA*$(\mathcal{A}, \mathcal{B})$ *for all subspace clusterings* $\mathcal{A}$*,* $\mathcal{B}$*.*

Proof.

$$\begin{aligned}\mathrm{RNIA}(\mathcal{A}^U, \mathcal{B}^U) &= \frac{(|U| + k) - |I|}{(|U| + k)} \\ &\geq \frac{|U| - |I|}{|U|} \\ &= \mathrm{RNIA}(\mathcal{A}, \mathcal{B}).\end{aligned}$$

□

Example 3. *(Penalty for Non-Intersecting Area with VI.) VI does not always penalize for non-intersecting area.*

We show by counterexample that VI does not always penalize for the non-intersecting area. Consider the clusterings in Example 1. We have the same intersection size but larger non-intersecting area for clusterings $\mathcal{B}$ and $\mathcal{C}$ than for clusterings $\mathcal{B}$ and $\mathcal{A}$, so $\mathcal{B}$ and $\mathcal{C}$ should be farther apart. Despite this, $\mathrm{VI}(\mathcal{B}, \mathcal{C}) < \mathrm{VI}(\mathcal{A}, \mathcal{B})$. This is true also if we scale the VI distances by the logarithms of the appropriate union areas.

Example 4. *(Penalty for Non-Intersecting Area with Rand.) The Rand index does not always penalize for non-intersecting area.*

Consider the clusterings from Example 2. With similar reasoning as in Example 3, the fact that $1 - \mathrm{Rand}(\mathcal{B}, \mathcal{C}) < 1 - \mathrm{Rand}(\mathcal{A}, \mathcal{B})$ shows that (1-Rand) fails to penalize for the growing non-intersecting area.

A.1.4 Scale Invariance

Consider scaling all areas by a constant $c \geq 1$.

Theorem 6. *(Scale Invariance for CE.) CE$(c\mathcal{A}, c\mathcal{B})$ = CE$(\mathcal{A}, \mathcal{B})$ for all subspace clusterings $\mathcal{A}$, $\mathcal{B}$.*

Proof.

$$\begin{aligned}\mathrm{CE}(c\mathcal{A}, c\mathcal{B}) &= \frac{c|U| - cD_{max}}{c|U|} \\ &= \frac{|U| - D_{max}}{|U|} \\ &= \mathrm{CE}(\mathcal{A}, \mathcal{B}).\end{aligned}$$

□

Theorem 7. *(Scale Invariance for RNIA.) RNIA$(c\mathcal{A}, c\mathcal{B})$ = RNIA$(\mathcal{A}, \mathcal{B})$ for all subspace clusterings $\mathcal{A}$, $\mathcal{B}$.*

Proof.

$$\begin{aligned}\mathrm{RNIA}(c\mathcal{A}, c\mathcal{B}) &= \frac{c|U| - c|I|}{c|U|} \\ &= \frac{|U| - |I|}{|U|} \\ &= \mathrm{RNIA}(\mathcal{A}, \mathcal{B}).\end{aligned}$$

□

Example 5. *(Scale Invariance for VI.) VI is not scale invariant in the case of subspace clusterings.*

We show this by counterexample.

$$\begin{aligned}&\mathrm{VI}(c\mathcal{A}, c\mathcal{B}) \\ &= \frac{1}{c|U|}\left(\sum_i (cm_i)\log(cm_i) + \sum_j (cm'_j)\log(cm'_j) - 2\sum_i\sum_j (cm_{ij})\log(cm_{ij})\right) \\ &= \mathrm{VI}(\mathcal{A}, \mathcal{B}) + \log c\left(\sum_i m_i + \sum_j m'_j - 2\sum_i\sum_j m_{ij}\right) \\ &\geq \mathrm{VI}(\mathcal{A}, \mathcal{B}),\end{aligned}$$

since for subspace clusterings, $\sum_i m_i \geq \sum_i \sum_j m_{ij}$ and $\sum_j m'_j \geq \sum_i \sum_j m_{ij}$. The equality (and thus scale invariance) only holds for ordinary clusterings, for which $m_i = \sum_j m_{ij}$ and $m'_j = \sum_i m_{ij} \quad \forall\, i, j$.

Example 6. *(Scale Invariance for Rand.) The Rand index is not scale invariant in the case of subspace clusterings.*

We show this by counterexample.

$$\begin{aligned}
1 - \text{Rand}(c\mathcal{A}, c\mathcal{B}) &= \frac{\sum_{i=1}^{L'} \sum_{j=1}^{L} \sum_{k=j+1}^{L} (cm_{ij})(cm_{ik})}{(cm)((cm)-1)/2} \\
&= \frac{\sum_{i=1}^{L} \sum_{j=1}^{L'} \sum_{k=j+1}^{L'} (cm_{ji})(cm_{ki})}{(cm)((cm)-1)/2} \\
&= \frac{\sum_{i=1}^{L'} \sum_{j=1}^{L} \sum_{k=j+1}^{L} m_{ij}m_{ik} + \sum_{i=1}^{L} \sum_{j=1}^{L'} \sum_{k=j+1}^{L'} m_{ji}m_{ki}}{m(m-1/c)/2} \\
&\leq 1 - \text{Rand}(\mathcal{A}, \mathcal{B}).
\end{aligned}$$

The equality holds only for $c = 1$, i.e. when there is no scaling.

A.1.5 Copy Invariance

Consider introducing two disjoint copies of the same clustering $\mathcal{S}$ in a large data matrix. Denote the new 'double clustering' by $\mathcal{S}^D$.

Theorem 8. *(Copy Invariance for CE.) CE$(\mathcal{A}^D, \mathcal{B}^D)$ = CE$(\mathcal{A}, \mathcal{B})$ for all subspace clusterings $\mathcal{A}$, $\mathcal{B}$.*

Proof.

$$\begin{aligned}
\text{CE}(\mathcal{A}^D, \mathcal{B}^D) &= \frac{2|U| - 2D_{max}}{2|U|} \\
&= \frac{|U| - D_{max}}{|U|} \\
&= \text{CE}(\mathcal{A}, \mathcal{B}).
\end{aligned}$$

□

Theorem 9. *(Copy Invariance for RNIA.) RNIA$(\mathcal{A}^D, \mathcal{B}^D)$ = RNIA$(\mathcal{A}, \mathcal{B})$ for all subspace clusterings $\mathcal{A}$, $\mathcal{B}$.*

Proof.

$$\begin{aligned}
\text{RNIA}(\mathcal{A}^D, \mathcal{B}^D) &= \frac{2|U| - 2|I|}{2|U|} \\
&= \frac{|U| - |I|}{|U|} \\
&= \text{RNIA}(\mathcal{A}, \mathcal{B}).
\end{aligned}$$

□

Theorem 10. *(Copy Invariance for VI.) VI($\mathcal{A}^D, \mathcal{B}^D$) = VI($\mathcal{A}, \mathcal{B}$) for all subspace clusterings $\mathcal{A}$, $\mathcal{B}$.*

Proof.

$$\begin{aligned}
&\mathrm{VI}(\mathcal{A}^D, \mathcal{B}^D) \\
&\quad = \frac{1}{2|U|}\left(2\sum_i m_i \log m_i + 2\sum_j m'_j \log m'_j - 2\cdot 2\sum_i\sum_j m_{ij}\log m_{ij}\right) \\
&\quad = \mathrm{VI}(\mathcal{A}, \mathcal{B}).
\end{aligned}$$

□

Example 7. *(Copy Invariance for Rand.) The Rand index is not copy invariant.*

We show this by counterexample. Note that (1-Rand) can also be written as

$$\begin{aligned}
1-\mathrm{Rand}(\mathcal{A},\mathcal{B}) = &\frac{1/2(\sum_{i=1}^{L'}\sum_{j=1}^{L}\sum_{k=1}^{L} m_{ij}m_{ik} - \sum_{i=1}^{L}\sum_{j=1}^{L'} m_{ij}^2)}{m(m-1)/2} \\
&+ \frac{1/2(\sum_{i=1}^{L}\sum_{j=1}^{L'}\sum_{k=1}^{L'} m_{ji}m_{ki} - \sum_{i=1}^{L}\sum_{j=1}^{L'} m_{ij}^2)}{m(m-1)/2}
\end{aligned}$$

Using this, we have

$$\begin{aligned}
1-\mathrm{Rand}(\mathcal{A}^D,\mathcal{B}^D) = &\frac{1/2(2\sum_{i=1}^{L'}2\sum_{j=1}^{L}2\sum_{k=1}^{L} m_{ij}m_{ik} - 2\sum_{i=1}^{L}2\sum_{j=1}^{L'} m_{ij}^2)}{(2m)((2m)-1)/2} \\
&+ \frac{1/2(2\sum_{i=1}^{L}2\sum_{j=1}^{L'}2\sum_{k=1}^{L'} m_{ji}m_{ki} - 2\sum_{i=1}^{L}2\sum_{j=1}^{L'} m_{ij}^2)}{(2m)((2m)-1)/2} \\
&= \frac{1/2(2\sum_{i=1}^{L'}\sum_{j=1}^{L}\sum_{k=1}^{L} m_{ij}m_{ik} - \sum_{i=1}^{L}\sum_{j=1}^{L'} m_{ij}^2)}{m(m-1/2)/2} \\
&+ \frac{1/2(2\sum_{i=1}^{L}\sum_{j=1}^{L'}\sum_{k=1}^{L'} m_{ji}m_{ki} - \sum_{i=1}^{L}\sum_{j=1}^{L'} m_{ij}^2)}{m(m-1/2)/2} \\
&\neq 1-\mathrm{Rand}(\mathcal{A},\mathcal{B})
\end{aligned}$$

in the general case.

A.1.6 Multiple Cluster Coverage Penalty

Consider two subspace clusterings $\mathcal{A} = (A_1)$ and $\mathcal{B} = (B_1, \ldots, B_K)$ such that the clusters B_i are disjoint, of equal size $|A_1|/K = |U|/K$, and fully cover $\operatorname{supp} A_1$.

Theorem 11. *(Multiple Cluster Coverage Penalty for CE.)* $CE(\mathcal{A}, \mathcal{B}) = \frac{K-1}{K}|U|$.

Proof. We allow only one of the clusters B_i to be matched with A_1, so $K-1$ clusters are left unmatched, and the clustering error becomes $\mathrm{CE}(\mathcal{A}, \mathcal{B}) = \frac{K-1}{K}|U|$. □

Theorem 12. *(Multiple Cluster Coverage Penalty for RNIA.)* $RNIA(\mathcal{A}, \mathcal{B}) = 0$.

Proof. In this case $I = U$ and therefore $\mathrm{RNIA}(\mathcal{A}, \mathcal{B}) = 0$. □

Theorem 13. *(Multiple Cluster Coverage Penalty for VI.)* $VI(\mathcal{A}, \mathcal{B}) = \log K$.

Proof. In this case, $M = (m_{1j})$ for $j = 1, \ldots, K$, where $m_{1j} = |U|/K$. Also, $m_1 = |U|$ and $m'_j = |U|/K$. We thus get

$$\begin{aligned}\mathrm{VI}(\mathcal{A}, \mathcal{B}) &= \frac{1}{|U|}\left[|U| \log |U| + K(|U|/K) \log(|U|/K) - 2K(|U|/K) \log(|U|/K)\right] \\ &= \log K.\end{aligned}$$

□

Theorem 14. *(Multiple Cluster Coverage Penalty for Rand.)* $1-Rand(\mathcal{A}, \mathcal{B}) = [|U|(K-1)][K(|U|-1)]$.

Proof. In this case, $N = |U|(|U|-1)/2$, $N_{11} = 1/2(|U|/K)(|U|/K-1)K$, and $N_{00} = 0$, so (1-Rand) becomes

$$\begin{aligned}1 - \mathrm{Rand}(\mathcal{A}, \mathcal{B}) &= 1 - \frac{N_{00} + N_{11}}{N} \\ &= 1 - \frac{1/2(|U|/K)(|U|/K-1)K}{|U|(|U|-1)/2} \\ &= \frac{|U|(K-1)}{K(|U|-1)}.\end{aligned}$$

□

A.1.7 Generalizability

In the case of non-axis-aligned subspace clusterings or attribute weighted clusterings, we are able to compute cluster sizes, unions, and intersections. However, since the cluster intersections are not integer-valued in general, we cannot compute the confusion matrix. That is, we cannot simply add extra singleton clusters with unit masses, since we do not have a way of handling the remaining non-integer mass.

If we have cluster sizes, unions, and intersections, but do not have a confusion matrix, are we still able to apply our distance measures? If we are, we call the distance measures generalizable.

Property 1. *(Generalizability for CE) CE is generalizable.*

Computing CE$(\mathcal{A}, \mathcal{B})$ according to Eq. 4.1 only requires cluster intersection sizes $|A_i \cap B_j| = m_{ij}$ and the union size $|U|$, so it follows that CE is generalizable.

Property 2. *(Generalizability for RNIA) RNIA is generalizable.*

Computing RNIA$(\mathcal{A}, \mathcal{B})$ according to Eq. 4.2 only requires the union size $|U|$ and the intersection size $|I|$ of the two clusterings, so RNIA is generalizable.

Property 3. *(Generalizability for VI) VI is generalizable.*

Computing VI$(\mathcal{A}, \mathcal{B})$ according to Eq. A.1 only requires cluster intersection sizes $|A_i \cap B_j| = m_{ij}$ and the union size $|U|$, so VI is generalizable.

Property 4. *(Generalizability for Rand) The Rand index is not generalizable.*

To compute (1-Rand), we need the confusion matrix of the two clusterings, so Rand is not generalizable.

A.2 Appendix 2: Proof for Theorem 1 of Section 5

Proposition 7. *If (u_j, v_j) is a principal pair, then $\sigma_j u_j = \Pi_{\mathcal{F}} v_j$ where $\Pi_{\mathcal{F}} x$ represents the projection of vector x on the subspace $\mathcal{F}$.*

Proof. (Proposition 7) True for $j = 1$ by the definition of the projection operation. For $j > 1$, we have by the definition of (u_j, v_j) that $u_j = \sigma_j^{-1} \Pi_{(u_1, \ldots u_{j-1})^{\perp_{\mathcal{F}}}} v_j$ where $(u_1, \ldots u_{j-1})^{\perp_{\mathcal{F}}}$ represents the orthogonal complement of $(u_1, \ldots u_{j-1})$ in $\mathcal{F}$. By the elementary "3 perpendicular theorem" we also have that $v_j \perp u_{j'}$ for any $j' < j$. Therefore, $\Pi_{(u_1, \ldots u_{j-1})} v_j = 0$ and $\Pi_{\mathcal{F}} v_j = \sigma_j u_j$. □

Proposition 8. *If U is a $n \times a$ matrix of rank at most p, then $||U||_F^2 \leq p||U||_2^2$, where $||\ ||_F$ represents the Frobenius norm.*

Proof. (Proposition 8) $||U||_F^2 = \operatorname{tr} U^T U \overset{1}{=} \sum_{j=1}^{p} \lambda_i(U^T U) \leq p\lambda_{max}(U^T U) = p||U||_2^2$. Equality $\overset{1}{=}$ holds because at most p eigenvalues of $U^T U$ are nonzero. □

Proof. (Theorem 1) Let $\sigma_{ij} = \cos(\theta^j_{\mathcal{F},\mathcal{G}_i})$ and let (u_{ij}, v_{ij}), $u_{ij} \in \mathcal{F}$, $v_{ij} \in \mathcal{G}_i$, $i = 1, \dots K$, $j = 1, \dots a_i$ be the principal vectors. Note that $\{v_{ij}\}$ form an orthonormal system and denote by V the $n \times a$ matrix $V = [v_{11}\, v_{12}\, \dots v_{Ka_K}]$. Define $\tilde{u}_{ij} = \Pi_{\mathcal{F}} v_{ij}$; $\tilde{u}_{ij}$ is a vector of length σ_{ij} with the same direction as u_{ij} by Proposition 7. Form the matrix U having $\tilde{u}_{ij}$, $i = 1, \dots K$, $j = 1, \dots a_i$ as columns. Then $||U||_F^2 = \mathrm{tr} U^T U = \sum_{ij} ||\tilde{u}_{ij}||^2 = \sum_{ij} \sigma_{ij}^2$.

It remains to show that $||U||_F^2 \leq p$. But, in matrix notation, $U = HV$ where H is the symmetric, idempotent ($H^2 = H$) matrix representing the projection onto $\mathcal{F}$. It is easy to verify that $||V||_2 = 1$ and $||H||_2 = 1$. Therefore, $||U||_2 \leq ||H||_2 ||V||_2 = 1$ and by virtue of Proposition 8 we obtain $||U||_F^2 \leq p$. □

A.3 Appendix 3: Proofs for Section 5.2

Consider two co-clusterings $\mathcal{S} = (S_{ij})$, $\mathcal{S}' = (S'_{ij})$ together with the corresponding row clusterings $\mathcal{R} = (R_i)$, $\mathcal{R}' = (R'_i)$ and the column clusterings $\mathcal{C} = (C_i)$, $\mathcal{C}' = (C'_i)$. For the row clusterings, we define the cluster sizes as $r_i = |R_i|$, $r'_i = |R'_i|$, and the cluster intersection sizes as $r_{ij} = |R_i \cap R'_j|$. Similarly, for the column clusterings, we have $c_i = |C_i|$, $c'_i = |C'_i|$, and $c_{ij} = |C_i \cap C'_j|$. For the co-clusterings, the sizes of the clusters are defined as $m_{ij} = |S_{ij}| = r_i c_j$, $m'_{ij} = |S'_{ij}| = r'_i c'_j$, and the intersection of two co-clusters is $m_{ijkl} = |S_{ij} \cap S'_{kl}| = r_{ik} c_{kl}$. Recall that m and p stand for the number of data matrix rows and columns, respectively. Also recall that D_{max} is the sum of the diagonal elements of the co-clustering confusion matrix after an optimal permutation of the co-cluster labels. Let us write D^R_{max} for the corresponding sum for the row clustering confusion matrix and D^C_{max} for the column clustering confusion matrix.

Theorem 15. *(CE and Co-Clusterings)* $CE(\mathcal{S}, \mathcal{S}') = CE(\mathcal{R}, \mathcal{R}') + CE(\mathcal{C}, \mathcal{C}') - CE(\mathcal{R}, \mathcal{R}') CE(\mathcal{C}, \mathcal{C}')$ *for any co-clusterings* $\mathcal{S} = (\mathcal{R}, \mathcal{C})$, $\mathcal{S}' = (\mathcal{R}', \mathcal{C}')$.

Proof. Fix the permutation of the co-cluster labels that minimizes $\mathrm{CE}(\mathcal{S}, \mathcal{S}')$.

The same permutation of the labels also minimizes $\mathrm{CE}(\mathcal{R}, \mathcal{R}')$ and $\mathrm{CE}(\mathcal{C}, \mathcal{C}')$.

$$\begin{aligned}
\mathrm{CE}(\mathcal{S}, \mathcal{S}') &= \frac{mp - D_{max}}{mp} \\
&= \frac{mp - \sum_i \sum_j m_{ijij}}{mp} \\
&= \frac{mp - \sum_i \sum_j r_{ii} c_{jj}}{mp} \\
&= \frac{mp - \sum_i r_{ii} \sum_j c_{jj}}{mp} \\
&= \frac{mp - D^R_{max} D^C_{max}}{mp} \\
&= \frac{m - D^R_{max}}{m} + \frac{p - D^C_{max}}{p} + \frac{(m - D^R_{max})(p - D^C_{max})}{mp} \\
&= \mathrm{CE}(\mathcal{R}, \mathcal{R}') + \mathrm{CE}(\mathcal{C}, \mathcal{C}') - \mathrm{CE}(\mathcal{R}, \mathcal{R}')\mathrm{CE}(\mathcal{C}, \mathcal{C}').
\end{aligned}$$

□

Theorem 16. *(RNIA and Co-Clusterings) RNIA*$(\mathcal{S}, \mathcal{S}') = 0$ *for any co-clusterings* $\mathcal{S} = (\mathcal{R}, \mathcal{C})$, $\mathcal{S}' = (\mathcal{R}', \mathcal{C}')$.

Proof. For co-clusterings it always holds that $I = U$. □

Theorem 17. *(VI and Co-Clusterings) VI*$(\mathcal{S}, \mathcal{S}') = VI(\mathcal{R}, \mathcal{R}') + VI(\mathcal{C}, \mathcal{C}')$ *for any co-clusterings* $\mathcal{S} = (\mathcal{R}, \mathcal{C})$, $\mathcal{S}' = (\mathcal{R}', \mathcal{C}')$.

Proof.

$$\begin{aligned}
\mathrm{VI}(\mathcal{S}, \mathcal{S}') &= \frac{1}{mp} \sum_i \sum_j \sum_k \sum_l m_{ijkl} \log \frac{m_{ij} m'_{kl}}{m^2_{ijkl}} \\
&= \frac{1}{mp} \sum_i \sum_j \sum_k \sum_l r_{ik} c_{jl} \log \frac{r_i c_j r'_k c'_l}{r^2_{ik} c'^2_{jl}} \\
&= \frac{1}{mp} \sum_i \sum_k r_{ik} \log \frac{r_i r'_k}{r^2_{ik}} \sum_j \sum_l c_{jl} \\
&+ \frac{1}{mp} \sum_j \sum_l c_{jl} \log \frac{c_j c'_l}{c^2_{jl}} \sum_i \sum_k r_{ik} \\
&= \frac{1}{m} \sum_i \sum_k r_{ik} \log \frac{r_i r'_k}{r^2_{ik}} + \frac{1}{p} \sum_j \sum_l c_{jl} \log \frac{c_j c'_l}{c^2_{jl}} \\
&= \mathrm{VI}(\mathcal{R}, \mathcal{R}') + \mathrm{VI}(\mathcal{C}, \mathcal{C}').
\end{aligned}$$

We get this by noticing that $\sum_j \sum_l c_{jl} = p$ and that $\sum_i \sum_k r_{ik} = m$. □

It does not seem possible to derive a simple relationship between the Rand index for the co-clusterings and the Rand indices for the corresponding row/column clustering. However, we are able to observe the following connection.

Based on Eq. A.3, we have the following expression for the row clusterings:

$$1 - \text{Rand}(\mathcal{R}, \mathcal{R}') = \frac{\sum_i r_i^2 + \sum_k r_k'^2 - 2\sum_i \sum_k r_{ik}^2}{m(m-1)}.$$

For the column clusterings, we have

$$1 - \text{Rand}(\mathcal{C}, \mathcal{C}') = \frac{\sum_j c_j^2 + \sum_l c_l'^2 - 2\sum_j \sum_l c_{jl}^2}{p(p-1)},$$

and for the co-clusterings,

$$\begin{aligned} 1 - \text{Rand}(\mathcal{S}, \mathcal{S}') &= \frac{\sum_i \sum_j m_{ij}^2 + \sum_k \sum_l m_{kl}^2 - 2\sum_i \sum_j \sum_k \sum_l m_{ijkl}^2}{m(m-1)} \\ &= \frac{\sum_i \sum_j r_i^2 c_j^2 + \sum_k \sum_l r_k'^2 c_l'^2 - 2\sum_i \sum_j \sum_k \sum_l r_{ik}^2 r_{jl}^2}{mp(mp-1)} \\ &= \frac{\sum_i r_i^2 \sum_j c_j^2 + \sum_k r_k'^2 \sum_l c_l'^2 - 2\sum_i \sum_k r_{ik}^2 \sum_j \sum_l c_{jl}^2}{mp(mp-1)}. \end{aligned}$$