Abstract

Clustering is one of the fundamental tasks of data mining. However, it is ill defined. There is no single definition of what a cluster is, and consecutively, an objective way to define the quality of a cluster. Nevertheless, traditional clustering methods produce a single solution, while data can be interpreted in many different ways and alternative clusterings may exist. In this project we define a taxonomy and present an overview of the existing alternative clustering methods. In addition, we have developed and implemented a new approach which extends the work presented in [De 11b], as part of the information theoretic framework for data mining [De 11a], which is based on the idea of the subjective interestingness of a clustering.

Acknowledgements

I would like to express my deepest appreciation and gratitude to my supervisor Tijl de Bie for his guidance and his patience throughout this project.

I'm also grateful to Professor Michael Vrahatis for the inspiration he gave me to continue my studies.

Finally, words alone cannot express the thanks I owe to my mother, Vassiliki, for making everything possible for me.

Contents

List of Figures

Chapter 1

Introduction

"Where is the Life we have lost in living?
Where is the wisdom we have lost in knowledge?
Where is the knowledge we have lost in information?"

T.S. Eliot

We are living in the Information age. The human kind had never before access to so much information so easily and so universally. The evolution of computers and more specifically the Internet, have played the central role in this spreading of information, providing the tools to transfer and store huge amounts of data. It is that information overload phenomenon and the thirst of human for knowledge that make data mining one of the most crucial fields in science. Knowledge discovery, defined by [FPSS+96] as *"the non- trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"* is more important than ever.

One of the most effective ways to extract knowledge from data is by grouping it. Cluster analysis has been a fundamental and widely used task in data mining for discovering patterns, relationships and structures in an unsupervised manner. The

problem lies in the very definition of clustering as a task; there is no objective way to define the quality of the clustering. Meanwhile, all the traditional clustering algorithms produce a single solution. We believe that different clusterings of the same data may exist and may be equally informative to different users. This idea is the motivation between the field of alternative clustering. In this context, there is no single solution for a given set of data, i.e. no optimal clustering; different clusterings emerge from different views of the same data.

Although alternative clustering has been a new field of research, there are numerous different approaches from many different viewpoints which have appeared in the literature lately. These approaches usually differ in their task formulation and their mathematical perspective. At the same time, there exists no overview of these methods, at least not to our knowledge. Thus, we have conducted a survey of the most important of them and a categorisation of is presented. This, is by itself important as it provides a solid source of information and references to the data miner and a basis for future, more updated and detailed surveys.

Furthermore, alternative clustering arises yet another question: *how can we quantify the* interestingness *of a clustering?* We believe that there is no objective way to define *interestingness*; it is a notion highly subjective to the practitioner. Thus, we bring into the picture of the data mining process the user, in a role equally important with that of the data. Based on the data mining framework [De 11a] we regard data mining as the information exchange between the data and the user, through a data mining algorithm and our ultimate goal is to update user's state of mind about the data.

This project aims at creating a method that that will reduce user's uncertainty about the data using clustering. In order to do that as effectively as possible, we provide her with different clusterings, each of which is a different view of the data, in an iterative manner, one by one. From all the possible clusterings, we choose the interesting to be presented to the practitioner. Hence, we define a quality metric that quantifies

the notion of *interestingness* for a clustering. The process begins with taking into account user's prior beliefs about the data and continues iteratively. In each iteration her updated beliefs are incorporated in our model allowing us to search for the most interesting clustering given the user's current state of mind.

We define all the aspects of our approach theoretically and we conduct experiments to prove the validation of our method. Most of the source code used for these experiments is available in B. In A the reader can find the mathematical preliminaries used for this work.

1.1 Aims and objectives

The core aims of this dissertation are **two**:

The **first**, to present a coherent overview of the approaches in alternative clustering that will be a source of useful information to any practitioner of the field. This part is important since its the first try, to our knowledge, in the field.

The **second**, to define our own approach. We are presenting a new method that takes into account not only the data but also the user and her beliefs in order to achieve a data mining task via clustering. This involves building a new theoretical model, extending the work in [De 11b] and performing experiments to argue about the validation of our approach.

The *first* aim's objectives are:

Decide on an appropriate taxonomy

A categorisation that would better capture the main differences of the methods and cluster them based on that is essential.

Choose the representative methods for each category

For each class of methods, the most important ones should be chosen based on their impact.

The *second* aim's objectives are:

Specify syntax of a pattern

We will define the syntax of a *clustering* pattern based on constraints on certain statistics of the dataset.

Search for the most interesting pattern

The hardest part of the method, since we will present an approximate method based on spectral clustering in order to solve (provably good) an NP-hard problem.

Self-information of a pattern

Define the quality metric we will use for the patterns in our approach.

Build an iterative model

Define how this model will work so that in each iteration we'll take into account the previously found clusterings and search for the next most informative one given these clusterings.

Validation

Argue in favour of our quality metric.

Conduct experiments

Test our method on synthetic data in order to extract useful insights.

1.2 Outline

This dissertation consists of 2 main parts. A survey of alternative clustering approaches and our approach in the field.

In Chapter 2, we present a general introduction to clustering. We briefly discuss what a proximity measure is, we present an overview of clustering methods as well as one of the most commonly used clustering algorithms; k-means. Finally, we introduce the reader to the field of alternative clustering.

In Chapter 3, we present an in depth survey on most of the major alternative clustering approaches, we define a taxonomy of these methods and discuss their core assets.

Chapter 4, is about our own approach to alternative clustering. The data mining framework, of which our approach is a special case, is overviewed. Most importantly, the core components of our approach, such as the prior beliefs of the user, the definition and the self information of a clustering pattern and finally the alternative clustering setting, are presented.

Chapter 5, presents experimental results of our project. Specifically, we argue about the validation our quality metric and we present results based on synthetic data.

Chapter 6, present our conclusions from the whole project as well as recommendations for future directions.

Chapter 2

Background and general context

"Essentially, all models are wrong, but some are useful."

George E. P. Box

2.1 Clustering

If someone is given a number of different objects and is asked to group them, it's a fairly easy process. But if then we ask her to describe how she did it, the answer is not trivial and will vary. As people interact with data they try to find features that describe it according to their views about it, compare it and group it based on the underlying notion of similarity. Clustering is a natural concept and it is based on subjective views on the data. However, it is ill defined in the sense that there exists no strict definition of what a cluster is.

Trying to describe it, we could say that it is the procedure of grouping a collection of unlabelled objects into subsets or "clusters" so that the objects assigned to the same cluster are more closely related or more similar to one another than objects assigned

to different clusters. This description makes it clear why there is a lack of strict definition; objects can not be grouped into clusters with a sole purpose in mind and not in a unique way. That is why the notion of "similarity" which will be generalised as "proximity" between objects is central for clustering.

In order to do any comparisons, we first need a *representation* of them, usually as abstract points in an d-dimensional space, depending on the number of their d features we are interested about. This should be the first step in any typical clustering process as described by [JD88] and [XW05]:

1. **Data representation and feature selection or extraction.** As pointed by [JMF99], data representation is about defining the number of data points and the number, type and scale of features. Then, via feature selection some distinguishing features will be selected, while some new ones will be generated via feature extraction [Bis95].

2. **Defining the suitable proximity measure.** A distance function must be used in order to quantify the proximity between the pairs of data points.

3. **Clustering.** This step can be utilised in various ways, depending on the choice of the clustering method. Essential for the clustering process is the **clustering criterion**, which as defined by [TK08], is an interpretation of what the practitioner defines as "sendible" based on what kind of clusters she expects to be underlying in the data and is usually presented in the form of a cost function.

4. **Clusters validation.** Since each clustering algorithm will eventually present a partitioning of the data, even if it is false or "meaningless" to the user, it is crucial to evaluate the results. The evaluation should be unbiased towards the clustering algorithms and provide the user with a degree of confidence.

5. **Data abstraction.** This is an optional step and it about choosing a simple and elegant way to interpret the results in order to provide meaningful insights about the data to the practitioner.

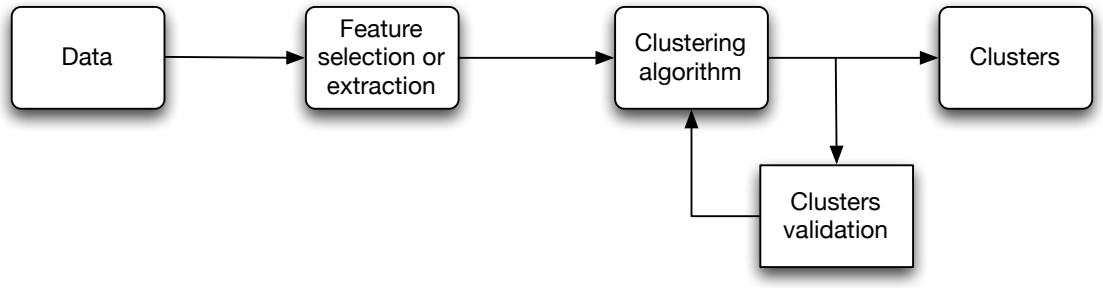


FIGURE 2.1: A typical clustering process. Data points are given as an input, features are selected or extracted, a proper pattern representation is selected, the clustering algorithm partitions the data based on proximity, the result may enter the validation loop and then clusters are revealed to the practitioner.

2.1.1 Proximity measures

In order to quantify *similarity*, the notion of **proximity measure** is introduced, as a way of quantifying how similar (or usually how different) two patterns are. Every clustering technique is trying to group the patterns based on a pre-defined proximity measure. Different proximity measures may produce different clusterings of the same data. That's why it is crucial for the clustering practitioner to be able to identify the right proximity measure for her purpose.

A common representation of proximity, is by proximity matrices which are given as an input to many clustering algorithms. Such a matrix for N objects is a symmetrical $N \times N$ matrix $\mathcal{D}$, with zero diagonal entries, where each entry d_{ij} has a record of the proximity between the i^{th} and j^{th} object, for $i, j = 1, \ldots, N$ and contains all the information about the proximities between all the pairs of the N patterns. This is so crucial for clustering that we could argue that clustering methods are just the way of summarising the information contained in that matrix in an understandable way for the data miner.

Definition 1. *Metric* is a function $d : X \times X \to \mathbf{R}$ such that $\forall x, y \in X$ it satisfies:

1. $d(x, y) \geq 0$ and $d(x, y) = 0 \Leftrightarrow x = y$ (positive definiteness)
2. $d(x, y) = d(y, x)$ (symmetric)
3. $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)

The most often setting for calculating the proximity between a pair of patterns x_i, x_j, in order to build the proximity matrix, is by using a distance (metric) to quantify the dissimilarity between them. The most common choice is the use of the *Euclidian* metric:

$$d_2(x_i, x_j) = \sqrt{\sum_{k=1}^{n}(x_{i,k}, x_{j,k})} = ||x_i - x_j||_2$$

as it provides an intuitive sense of distance for patterns in two or three dimensional spaces (n is the dimension of the space) and is a special case ($p = 2$) of the more general *Minkowski* metric:

$$d(x_i, x_j) = \big(\sum_{k=1}^{n}(x_{i,k}, x_{j,k})\big)^{\frac{1}{p}} = ||x_i - x_j||_p$$

Many times, features of the patterns may not be comparable in this way, so there also exist other ways of computing the proximity between two patterns e.g using different metrics, combinations of them or even completely different settings depending on the representation which is used, many of which are presented in [TK08].

2.1.2 Clustering methods

Based on the taxonomy presented by [JMF99], there are two main categories in clustering algorithms: **Hierarhical** and **Partitional** methods which are also divided in subcategories as seen in Figure 2.2.

Briefly, Hierarchical methods first organise the data into a nested sequence of groups, like a dendodiagram, so that the data miner is able to choose a clustering in a certain

level of proximity. Partitional methods on the other hand, select a clustering criterion and evaluate it for all possible partitions of the data containing a fixed number of K clusters. This however, is a difficult combinatorial problem and other, more optimised methods exist for Partitional clustering i.e. begin with an initial partition of the data and move objects as the clustering criterion function improves.

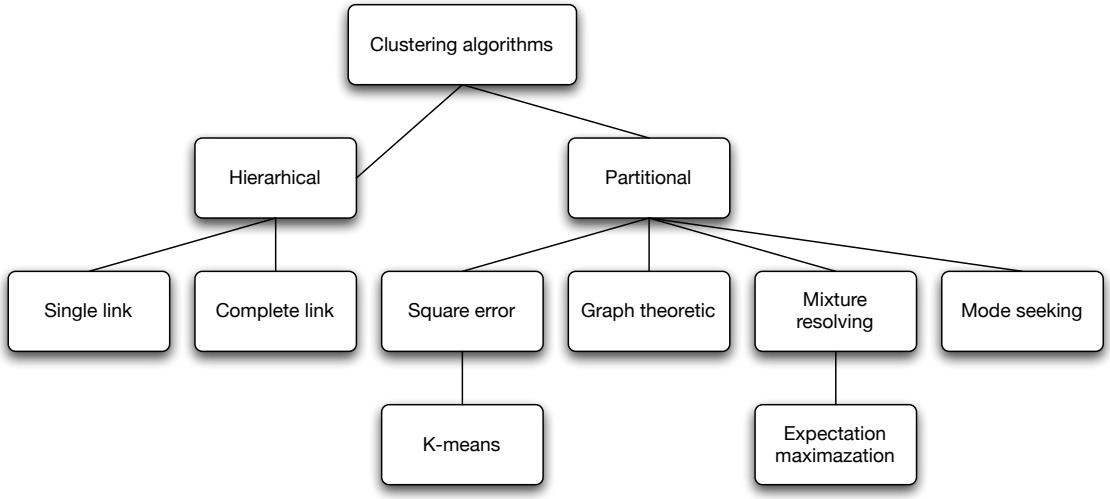


FIGURE 2.2: A taxonomy of clustering algorithms

In order for this taxonomy to be sound, we need to mention some of the other dichotomies which appear throughout all of its levels as they have been presented by [JMF99] and [JD88].

- *Agglomerative* versus *Divisive*: An agglomerative method begins with each object as cluster and keeps merging clusters while a divisive method goes the opposite way beginning with a single cluster and keeps splitting.
- *Monothetic* versus *Polythetic*: A monothetic method uses one feature at a time while polythetic uses all features at once.
- *Hard* versus *Fuzzy*: A hard method allows each object to be assigned in only one clustering while a fuzzy method uses degrees of membership for an object to several clusters.

Of course this is just a part of the existing categorisations of clustering methods, but we believe that these are the essential categories that the reader must keep in mind for the scope of this report. More about the clustering methods can be found on the surveys of [XW05] and [JMF99].

2.1.3 Clustering algorithm, an example: K-means

To fully understand how a clustering method works, we must get insight in an existing clustering algorithm, which in our case will be one of the most widely used; the K-means algorithm.

In this section we will present the K-means algorithm as a variant of the *Expectation-Maximisation* (EM) algorithm [DLR77] in the way it is described by [Bs06].

Given a data set of N data points $\{x_1, \ldots, x_N\}$ existing in an d-dimensional space, we want to partition them into K clusters, where K is a given number. We introduce the notation of **prototypes** μ_k, for $k = 1, \ldots, K$ as d-dimensional vectors associated with the k^{th} cluster which represent the centres of the clusters. To indicate in which clusters a data point x_n is assigned to, we introduce the notation of a set of indicator variables $r_{nk} \in \{0, 1\}$, for $k = 1, \ldots, K$. We also define the objective function J, also known as a *distortion measure* as

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||x_n - \mu_k||^2$$

We are aiming at finding a set of $\{r_{nk}\}$, an assignment of the N data points to the K clusters, and a set of vectors $\{\mu_k\}$ where J, the sum of the squares of the distances of each data point from its closest prototype μ_k, is minimum.

We can reach our goal through an iterative process consisting of two successive steps in each iteration. The first (*Expectation*) step optimises J with respect to $\{r_{nk}\}$ keeping

$\{\mu_k\}$ fixed, while then second (*Maximisation*) step optimises J with respect to $\{\mu_k\}$ keeping $\{r_{nk}\}$ fixed.

Algorithm 1 K-means

initialise the set of prototypes $\{\mu_k\}$ to random values.

while the assignments in the expectation step do not change **do**

expectation step. Each data point x_n is assigned to the closest centre.

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j ||x + n - \mu_j||^2 \\ 0 & otherwise \end{cases}$$

maximisation step. Update

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

end while

2.2 Alternative Clustering

As we stated in the previous chapter, Clustering is an ill-defined notion which is highly subjective. In other words, we are unable to define a "correct" clustering in an objective way. What is more, it has been proved by [Kle03] that for three fundamental properties of clustering: *scale-invariance*, *richness* and *consistency*, there exists no single clustering of the data that satisfies all of them.

For example, a user may be supplied with data about a group of people. She may already be aware of the clusters "women" and "men" or simply not interested in these, and she would prefer a clustering that would provide more knowledge. Another trivial example is shown in Figure 2.3, where there are two different clusterings for the same data, potentially with equal importance for the user.

One motivation behind alternative clustering, and our approach, is to take into account any prior knowledge the user may have about the data and use it to obtain new, more useful results and update her knowledge. The question arising is if there exist any other clusterings of the data that are potentially equally or more interesting to the user. This question is the main motivation for alternative clustering.

Other motivations are to provide user with options so she can discover what she wants even if initially she didn't know, provide different viewpoints for the data or sometimes even verify whether multiple explanations exist or not.

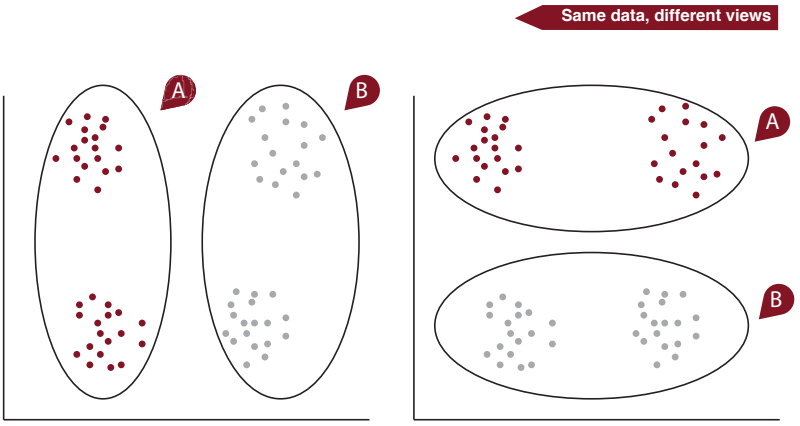


FIGURE 2.3: Two different ways to cluster the same data set.

A definition for the **alternative clustering problem** is given by [DQ08] and [QD09]: *"for a given objective function f and an existing clustering C such that $f(C) = x$ does there exist an alternative clustering C' such that $C \neq C'$ and $f(C) \approx f(C')$?"*

Or as given by [BB06]: *"Given a clustering C with r clusters, find a second clustering C' with r clusters, having high dissimilarity to C, but also satisfying a quality requirement threshold ω"*

But still, more questions arise. Questions that act as motivations for various aspects in alternative clustering.

- How much should C and C' overlap?
- Should they be totally different?
- Can *quality* be defined objectively?
- What is the trade-off between "alternativeness" and quality?

Chapter 3

Alternative clustering methods

"Science is what we understand well enough to explain to a computer.
Art is everything else we do."

Donald Knuth

Despite the fact that alternative clustering did not get much attention in the literature until recently, there is a rapidly growing number of different approaches.

One main difference in these approaches is in the task formulation. In our survey we will exploit this difference in order to categorise the different approaches into two main categories: *Semi-supervised* and *Unsupervised*. Here, the term "semi-supervised" is used to imply that these approaches take into account some kind of side information (e.g. cannot-link constraints, negative information) while "unsupervised" approaches use no a priori knowledge.

Since the unsupervised approaches use no a priori knowledge, they produce a number of clustering solutions simultaneously while the semi-supervised, based on existing knowledge produce their solutions in a sequential way.

- **Simultaneous generation of alternative clusterings:** Simultaneous generate a set of clusterings $\{C_1, \ldots, C_n\}$ such that each C_i, for $i = 1, \ldots, n$ is of *high quality* and $\forall(C_i, C_j), i \neq j \ C_i \neq C_j$.

- **Sequential generation of alternative clusterings:** Given a set of clusterings $\{C_1, \ldots, C_n\}$ generate a new clustering C_{n+1} such that C_{n+1} is of *high quality* and not similar to any of the previous n clusterings Repeat until a criterion is satisfied.

Another main difference, that will be significant enough to create a third category of approaches, is the style of the algorithm. Some of them are based on *data transformation*, projecting data into an orthogonal subspace and then perform clustering aiming for a new clustering solution, while others are based on creating a new *objective function* that will produce an alternative clustering trading-off quality and dissimilarity of the solution produced. It is obvious that the first category (data transformation) can only be part of the *semi-supervised* methods since it uses an existing clustering.

- **Data-transformation-oriented** approaches try to transform the data by projecting it into an orthogonal subspace so that the clustering algorithm is more likely to find novel features of the data and thus generate an alternative clustering.

- **Objective-function-oriented** approaches try to address the alternative clustering problem using different objective functions that take into account quality and dissimilarity of the alternative clustering(s) generated.

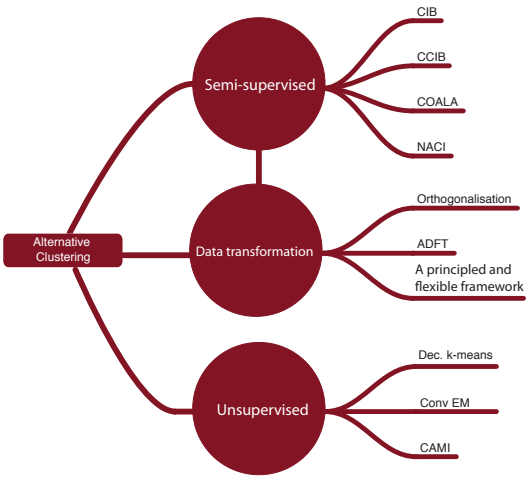


FIGURE 3.1: An illustration of the taxonomy for the alternative clustering methods

In conclusion, for the scope of this survey we will categorise the different approaches into three different categories that better reflect their differences:

1. Semi-supervised
2. Data transformation oriented
3. Unsupervised

and we will present the most representative approaches of each category.

3.1 Semi-Supervised

The approaches described in this category take into account an existing clustering solution as side information. Their sequential manner of producing solutions makes them greedy since in each step they choose to produce the optimal clustering in means of dissimilarity and quality with respect to some previous clustering solution.

The main drawback of this category of approaches is that many times clustering is used as the first step in a data mining task and no previous knowledge about the data is given.

In the context of sequential generation of alternative clustering, the alternative clustering problem can be defined [DQ08] [QD09] (and slightly different by [BB06]) as follows: *"for a given objective function f and an existing clustering C such that $f(C) = x$ does there exist an alternative clustering C' such that $C \neq C'$ and $f(C) \approx f(C')$?"*

3.1.1 Conditional information bottleneck

One of the first approaches to alternative clustering was as an application of the *Conditional Information Bottleneck (CIB)*, a framework proposed by [GH03] as a generalisation of the well known *Information Bottleneck method (IB)* [TPB00].

The *a priori* known structures or properties of the data are regarded as side information and the goal is to maximise the conditional information about the relevant structures, conditioning on side information. An application of this general framework is an alternative clustering setting, where a known categorisation of the data is seen as side information, and the goal is to obtain a new clustering solution.

Information bottleneck

In *Information bottleneck*, structure extraction is modelled as data compression and the relevance of the extracted structure is quantified by the info it preserves about a specified relevance variable. Given two variables X (i.e. objects) and Y (i.e. features) and their joint distribution, the shared information between these two variables is maximised while one is compressed through a third variable C (i.e. clusters). This

method proposes a probabilistic clustering setting, in which given a stochastic mapping $P(c|x)$ of objects to clusters and some side information as a random variable Z :

$$\min_{p(c|x)} F \equiv I(X,C) - \beta I(Y,C|Z)$$
$$s.t. \quad \sum_c p(c|x) = 1, \forall x, c \;\; and \;\; p(c|x) \geq 0$$

where β is a positive scalar that balances the trade-off between compression and preservation of information about the features of interest.

where

$$p(c|x) \propto q(c) \exp\big(\beta \sum_y p(y|x) \log q(y|c)\big) = p(c) \exp\big(-\beta D_{KL}(p(y|x)||p(y|c))\big)$$

and $q(c)$ and $q(y|c)$ are auxiliary parameters.

Conditional information bottleneck

In CIB, a new variable Z is introduced, representing the pre-defined class labels of the given clustering (as negative information) and the new objective is to find the optimal assignment of X to C while preserving as much information about Y conditioned on Z as possible. The new objective function is

$$\tilde{F} = -H(C|X) + \tilde{H}(C) + \beta \tilde{H}(Y|Z,C)$$

CIB is looking for a probabilistic assignment of data to clusters with minimal fuzziness and at the same time maximal information encoded jointly in variables C and Z.

$$P^*_{C|X} = \arg\max_{P_{C|X} \in \mathcal{P}} I(C;Y|Z)$$

$$\mathcal{P} \equiv \{P_{C|X} : I(C;X) \leq C_{max}\}$$

All in all, CIB is a relatively old and outdated approach which however, paved the way for future approaches and alternative clustering as a field. The major drawback of this method, is its prerequisite for a joint distribution, which is known to be hard to formalise.

3.1.2 Coordinated conditional information bottleneck

An extension to the previous framework (CIB 3.1.1) was presented by the same authors in [GH04], called *Coordinated Conditional Information Bottleneck (CCIB)*.

The authors base their new approach on a weakness of their previous framework, CIB. Since $I(C;Y,Z)$ measures the information conveyed by C and Z in conjunction, it lacks in presenting the amount of information C provides without knowledge of Z. Or as they call it *cluster coordination problem*.

They address this problem by using an additional constraint regarding $I(X;Y)$:

$$P^*_{C|X} = \arg\max_{P_{C|X} \in \mathcal{P}} I(C;Y|Z)$$

$$P \equiv \{P_{C|X} : I(C;X) \leq C_{max}, I(C;Y) \geq I_{min}\}$$

3.1.3 Conditional ensemble clustering

The third work in this line by Gondek and Hofmann [GH05], present a framework that is also driven by side information, but this time, it is about negative side information i.e. *undesired* clustering. They argue that it is usually easier for the user to define what she *is not* looking for rather than what she *is* looking for.

The framework, given a dataset $\{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$, a clustering $Z : \{1, \ldots, n\} \to \{1, \ldots, l\}$, the number k of target clustering and the number k_j for each local clustering, works in 3 stages:

1. **Compute local clusterings for each cluster of the given clustering**
   Partition data set into pre-image sets:
   $I_j(Z) \equiv \{i : Z(i) = j\}$ and $X_j(Z) \equiv \{x_i : i \in I_j(Z)\}, \quad j = 1, \ldots, l$
   Apply base clustering method to each $X_j(Z)$ to find a local clustering
   $\hat{C}^j : I_j(Z) \to \{1, \ldots, k_j\}, \quad j = 1, \ldots, l$

2. **Extend local clustering solutions to global clusterings**
   Extend each $\hat{C}^j$ to a global clustering C^j by assigning given instances in $X_m(Z)$, $m \neq j$ to one of the existing clusters
   $\hat{C}^j : \{1, \ldots, n\} \to \{1, \ldots, k_j\}, \quad j = 1, \ldots, l$

3. **Combine clustering solutions using an ensemble clustering**
   Clustering solutions C^j are combined to form the consensus clustering:
   $C = Consens(C^1, \ldots, C^l)$, where $C : \{1, \ldots, n\} \to \{1, \ldots, k\}$

An advantage of this framework is its flexibility to use different clustering methods on the second step. This, allows the user to employ a clustering algorithm according to the specific domain of the data.

3.1.4 COALA

Another technique, that falls into the category of semi-supervised alternative clustering, known as COALA [BB06], is based on optimising an objective function that combines the requirements for *disimilarity* and *quality* for the generated alternative clustering.

The first requirement, which tries to ensure that given a new clustering S as a solution, S is as dissimilar as possible from an already known clustering C, is addressed by using instance-based pairwise '*cannot-link*' constraints. The second requirement, which tries to ensure that a clustering that presented as solution is of high quality and depends on the distance function used, is addressed by a pre-specified quality threshold ω that plays the role of balancing the trade-off between the two requirements.

More specifically, COALA is based on an agglomerative hierarchical algorithm, using *average-linkage* [Voo86] as a distance function. The technique works in two steps. The first, can be seen as a preliminary process that will make use of the existing clustering to generate the constraints that will be used in the second step, where the alternative clustering will be generated.

Algorithm 2 Constraints generation

Require: clustering $C = \{c_1, \ldots, c_n\}$, constraints set $L = \{\}$
  for $i = 0$ to n **do**
    for $j = 0$ to $|c_i$ **do**
      for $k = j + 1$ to $|c_i$ **do**
        $L = L \cup addConstraint(x_j, x_k)$
      end for
    end for
  end for

The second step is that of a classical hierarchical algorithm, where n different clusters are being initially generated and then iteratively start merging. Here, COALA defines the candidates for this merging categorising the pairs of objects into *qualitative* $((q_1, q_2))$ and *dissimilar* $((o_1, o_2))$. The first, denotes the pairs with the smallest

distance and the second the pairs with the smallest distance conditioned on the constraints. The motivation behind generation of the dissimilar pairs, is the hope to satisfy the dissimilarity requirement for the overall clustering by building it (i.e. hierarchical agglomerative) by choosing dissimilar pairs.

However, the quality component isn't ignored by the technique, which uses the quality threshold ω to decide each time in favour of a qualitative or a dissimilar merging. This is done by comparing the ratio $\frac{d_q}{d_o}$ with ω. This reveals the crucial role that the user choice of ω plays in balancing the trade-off between quality and dissimilarity.

In order to quantify the dissimilarity, COALA uses the Jaccard Index [PM06] which measures the dissimilarity between two clusterings C and S as:

$$J(C,S) = \frac{C \cap S}{C \cup S} = \frac{M_{11}}{M_{11} + M_{01} + M_{10}}$$

where M_{11} is the number objects in the same cluster for both clusterings and M_{01}, M_{10} are the number of objects that exist in the same cluster for one clustering but not the other.

To measure quality, the Dunn Index [Dun73] for a clustering $C = \{c_1, \ldots, c_n\}$ is used:

$$DI(C) = \min_{i \leq n} \left\{ \min_{1 \leq j \leq n, i \neq j} \left\{ \frac{d(c_i, c_j)}{\max_{1 \leq k \leq n} d'(c_k)} \right\} \right\}$$

where $d(i,j)$ is the distance between clusters i and j, and $d'(k)$ is a cluster diameter measure for cluster k.

The authors also propose *DQ-Measure* [BB06], an overall measure for quality and dissimilarity between two clusterings C and S which combines the two measures stated above:

$$DQ(C,S) = \frac{2J(C,S)DI(C,S)}{J(C,S) + DI(C,S)}$$

3.1.5 NACI

A method called NACI (*Non-linear Alternative Clustering with Information theory*) is presented in [DB10b]. This method also tries to balance the trade-off between quality and dissimilarity of the generated alternative clustering via a dual objective function (similarly to COALA 3.1.4).

This technique tries to optimise a clustering objective function based on mutual information; the mutual dependence between C^+, the clustering labels, and X, the data observations. Intuitively, this states that given some data observations X we can infer the information about the values of the clustering labels with small error. Using Fano's inequality from information theory, which relates the average information lost to the probability of categorisation error:

$$P(c^+ \neq \widehat{c^+}) \geq \frac{H(C^+|X) - 1}{\log|C'|} = \frac{H(C^+) - I(C^+;X)}{\log|C^+|}$$

where $H(C^+|X)$ is the conditional entropy of C' given X, c^+ and $\widehat{c^+}$ represent the true and the guessed cluster labels of C' given X. This means that the lower bound provided by Fano's inequality about the error probability is minimised as the mutual information between C^+ and X is maximised. Hence, the dual objective function is defined as:

$$C' = \arg\max_{C^+}\{I(C^+;X) - \eta(C^+;C^-)\} \tag{3.1.1}$$

for some parameter η which balances the trade-off of dissimilarity and quality.

In order to optimise (3.1.1), NACI uses the classic agglomerative hierarchical method. However, instead of deciding the merging between two clusterings in C^+ based on their distance, this technique merges them only if such a merging results in an increased global mutual information $I(C^+; X)$ and a minimised $I(C^+; C^-)$.

3.2 Data transformation oriented

Data transformation oriented methods, are part of the semi-supervised methods, in the sense that they use an existing clustering as side-information in order to produce a new clustering solution. They distinguish themselves from the approaches of the previous section since they use this side-information in a completely different manner. These approaches transform data into new subspaces in order to get novel clusterings.

3.2.1 A framework using orthogonalisation

In [CFD07], the authors present a framework that includes two methods that use data projection of the data in order to obtain new, non-redundant clusterings. Intuitively, this framework proposes an initial clustering of the data, then a transformation of the data into an orthogonal space (which is not covered by the existing clustering) and iterate until most of the data space is covered.

The first method performs orthogonal clustering (seeks orthogonality in the cluster space) while the second performs clustering in orthogonal subspaces (seeks orthogonality in the feature subspace). The distinction between the two methods proposed is based on the representation of the given clusterings and consequently in the way the orthogonalisation of the data is achieved.

In the first method (3.2.1.1), a clustering solution is represented using the k centroids, while in the second method (3.2.1.1), using the feature subspace that best captures the clustering.

3.2.1.1 Orthogonal clustering

In this method, clustering is viewed as a way of compressing data $\mathcal{X} \in \mathbb{R}^{d \times N}$, where each data point $x_i \in \mathbb{R}^d$, $i = 1, \ldots, N$ is represented by its centroid μ_j, $j = 1, \ldots, k$. The space spanned by x_i is the data space while the space spanned by m_j is the *compressed* data space. Using this representation, the authors, project data points from the original data space to the compressed data space. What is not covered by the compressed data space is called the *residue* space, which the authors define as the space as the data projected onto the space orthogonal to a given representation. Hence, to find alternative clusterings, the method performs clustering in that space.

Given a clustering of the data $X^{(t)}$, $M^{(t)} = [\mu_1^{(t)}, \ldots, \mu_k]^{(t)}$, a representation in the residue space depends on the fuzziness of the membership of each data point to a cluster.

1. For *hard* clustering, where each data point x_i belongs only to cluster j, it is projected on centroid μ_j and the residue $x_i^{(t+1)}$ is the projection of $x_i^{(t)}$ to the subspace orthogonal to μ_j through the following equation:

$$x_i^{(t+1)} = \frac{\mathbf{I} - \mu_j^{(t)} \mu_j^{(t)\prime}}{\mu_j^{(t)\prime} \mu_j^{(t)}} x_i^{(t)}$$

2. For *soft* clustering, where each data point x_i has a degree of membership to all clusters, the data is projected to the space spanned by all the centroids of the clustering, and the residue space $X^{(t+1)}$ is found as the projection of $X^{(t)}$ to the

space orthogonal to all centroids using the following equation:

$$X^{(t+1)} = (\mathbf{I} - M^{(t)}(M^{(t)'}M^{(t)})^{-1}M^{(t)'})X^{(t)}$$

3.2.1.2 Clustering in orthogonal subspaces

In this method, the representation of a given clustering $M = [\mu_1, \ldots, \mu_k]$, is by finding the feature subspace that best captures the clustering structure. This can be achieved either by applying either Linear Discriminant Analysis (LDA) or Principal Component Analysis (PCA), both giving similar results.

After finding the feature subspace $A = [\phi_1, \ldots, \phi_{k-1}]$, the dataset $X^{(t)}$ is projected to a space orthogonal to A to obtain the residue $X^{(t+1)}$:

$$\begin{aligned} X^{(t+1)} &= P^{(t)}X^{(t)} \\ &= (\mathbf{I} - A^{(t)}(A^{(t)'}A^{(t)})^{-1}A^{(t)'})X^{(t)} \end{aligned}$$

3.2.2 Finding alternative clusterings using constraints

The approach presented in [DQ08], transforms data X into a new space X' either by applying $X' = D'^{T}X$ for some distance function D and lets the algorithm use its distance function e.g. the Euclidian, or the algorithm uses data X replacing with its distance function with D'. Given an initial clustering π of the data X, the whole process can be summarised in 3 steps:

Characterising step using constraints

Based on clustering π, extract a set of *must-link* or *cannot-link* constraints C,

and learn a distance function D_π from C. This can be achieved in a variety of ways [XNJR03].

Alternative calculation

Find an alternative distance function D'_π from D_π by applying a single value decomposition (SVD) to D_π, such that $D_\pi = HSA$, where H the hanger matrix, S the stretcher matrix and A the aligner matrix. Hence, $D'_\pi = HS^{-1}A$.

Transformation

Use D'_π to transform the data: $X' = D'^T_\pi X$.

Re-clustering

Perform clustering on the transformed data X'.

A big advantage of this approach is its **algorithm-independent** manner, which allows the user to choose the clustering method that better fits her needs.

3.2.3 A principled and flexible framework

The framework proposed in [QD09], by the same authors as the previous approach (3.2.2), which share the same **algorithm-independent** nature, suggests that the user should be able to define some of the properties of the existing (given) clustering into his new, alternative clustering solution. This means that the user is able to find a partially alternative clustering instead of a completely new one. This is achieved by creating a transformation matrix which transforms the data into a new space and at the same time it preserves the properties of the data while it takes into account the user's feedback on the previous clustering.

The method transforms a dataset $X = \{x_1, \ldots, x_n\}$, $X \in \mathbb{R}^{d \times n}$, with a given clustering π containing clusters $C_1, \ldots, C_k$ with centroids $[\mu_1, \ldots, \mu_k]$, through a transformation matrix $D \in \mathbb{R}^{d \times d}$ matrix to a transformed dataset $Y = DX$, $Y \in \mathbb{R}^{d \times n}$. The alternative clustering π' will be produced by applying any clustering algorithm on Y.

The formulation proposed minimises a Kullback-Leibler (A.1) divergence between the original (p_x) and the transformed data distribution (p_y), constrained on the properties of the previous clustering that should be kept:

$$\min_{B \geq 0} D_{KL}(p_Y(y)||p_x(x))$$

$$s.t. \ \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1, x_i \notin C_j}^{k} ||(x_i - \mu_j)||_B^{\alpha} \leq \beta$$

where B is the weight matrix $B = D'D$ and $|| \cdot ||$ denotes the Mahalanobis distance. An assumption being made is that each cluster in the alternative clustering π' follows a multivariate Gaussian distribution with the same covariance matrix Σ (A.1). The parameter $\alpha \geq 1$ defines the trade-off between dissimilarity and quality of the new clustering.

3.3 Unsupervised

Unsupervised (simultaneous creation of clustering solutions) approaches are non-greedy. They generate a globally optimal collection of n different clusterings of the data. Since no previous clusterings are given as an input, this kind of approaches are closer to the *unsupervised* nature of clustering. Their drawback, however, is that generating n alternative clusterings at once, may be a *much* harder optimisation problem.

3.3.1 De-correlated k-means and convolutional EM

De-correlated k-means

In this approach, proposed in [JMD08], an objective function is formulated that combines the error of a clustering an a *correlation* between clusterings. The authors introduce a measure of correlation between clusterings and present a k-means-style (2.1.3) algorithm for minimising it.

More specifically, given a dataset $X = \{x_1, \ldots, x_n\} \in \mathbb{R}^d$, the method partitions X into two clusters C^1 and C^2 with M_1 and M_2 groups respectively. In order to achieve this goal, the method tries to find *decorrelated* clusterings each of which is a complete partition of the dataset X.

The objective function that needs to be minimised is the following:

$$\begin{aligned} G(\mu_{1\ldots M_1}, \mu_{1,\ldots M_2}) = &\sum_i \sum_{z \in C_i^1} ||z - \mu_i||^2 \\ &+ \sum_j \sum_{z \in C_j^2} ||z - \nu_j||^2 \\ &+ \lambda \sum_{i,j} (\beta_{\mathbf{j}}{}' \mu_i)^2 \\ &+ \lambda \sum_{i,j} (\alpha_{\mathbf{i}}{}' \nu_i)^2 \end{aligned}$$

where C_i^1 the i^{th} cluster of the first clustering, C_j^2 the j^{th} cluster of the second clustering, $\lambda > 0$ a parameter that trades off the clustering error and decorrelation of the clusterings , μ_i is the representative vector for C_i^1 and ν_j the representative vector for C_j^2, $\alpha_{\mathbf{i}}$ the mean vector of C^1 and $\beta_{\mathbf{j}}$ the mean vector of C^2.

The first two terms of the objective function are about the clustering error, similar to the error type of k-means, with the difference that the representative vectors are not always the mean vectors. The other two terms measure the *decorrelation* between the two clusterings.

The approach initialises one of the two clusterings using k-means with $k = M_1$ and the other one randomly. To optimise this objective function iteratively, in each step μ_i and ν_j must be updated:

$$\mu_i = (I - \xi V Q(I + \xi_i \Sigma)^{-1} Q' V')\alpha_{\mathbf{i}}$$

$$\nu_j = (I - \zeta_j M U(I + \zeta_j \Lambda)^{-1} U' M')\beta_{\mathbf{j}}$$

where $xi_i = \frac{\lambda}{\sum_j n_{ij}}$, $V = [\beta_{\mathbf{1}}, \ldots, \beta_{\mathbf{M_2}}]$, $V'V = Q\Sigma Q'$ the eigenvalue decomposition, $\zeta_j = \frac{\lambda}{\sum_i n_{ij}}$, $M = [\alpha_{\mathbf{1}}, \ldots, \alpha_{\mathbf{M_1}}]$ and $M'M = U\Lambda U'$ the eigenvalue decomposition.

Convolutional EM

This method, also known as *Sum of parts approach*, was proposed together with the previous one in [JMD08], although it is in a complete different line except the part that they are both completely unsupervised. The main idea is that the data is modelled as a sum of independent mixtures models. Each of these independent components is associated with a clustering.

Extending the previous method's notation, a set of observations $\mathcal{Z} = \{x_1, \ldots, x_n\} \in \mathbb{R}^d$, sampled from a random variable Z, the method models Z as the summation of two random variables, X and Y.

The approach models the data as a sample of a convolution of two mixtures. Thus, it deals with the problem of learning a convolution of mixtures.

$$p_X = \sum_{i=1}^{M_1} \alpha_i p_{X_i}, \quad p_Y = \sum_{i=1}^{M_2} \beta_j p_{X_j}$$

$$p_Z(z) = (p_X \cdot p_Y)(z) = \sum_{i=1}^{M_1} \sum_{i=1}^{M_2} (\alpha_i \beta_j)(p_{X_i} p_{Y_j})$$

More specifically, given the data $\mathcal{Z}$, find the parameters of p_X, P_Y and the weights α_i, β_j. In order to do that, the authors propose an EM algorithm. Under the assumption that the X, Y are mixtures of spherical Gaussian distributions:

$$p_X = \sum_{i=1}^{M_1} \alpha_i \mathcal{N}(\mu_\mathbf{i}, \sigma^2), \quad p_Y = \sum_{i=1}^{M_2} \beta_j \mathcal{N}(\nu_\mathbf{i}, \sigma^2)$$

The algorithm is initialised by k-means for the first clustering and a random assignment for the second clustering, the vectors $\mu_\mathbf{i}^\mathbf{0}$ and $\nu_\mathbf{j}^\mathbf{0}$ to the means of the first and second clustering and finally $\sigma = \frac{1}{\sqrt{2m}}(\min_{i \neq j} ||\mu_\mathbf{i}^\mathbf{0} - \mu_\mathbf{j}^\mathbf{0}||, ||\nu_\mathbf{i}^\mathbf{0} - \nu_\mathbf{j}^\mathbf{0}||)$. Let $p_{ij}^t(z)$ denote the conditional probability that z comes from the normal distrubution $p_X \cdot p_Y$ given the current parameters.

E-step:

$$p_{ij}^{t+1}(z) = \begin{cases} 1, & if \quad (i,j) = \arg\max_(\, r,s)\{a_r^t b_s^t \cdot \mathcal{N}(\mu_\mathbf{r}^\mathbf{t} + \nu_\mathbf{s}^\mathbf{t}, 2('sigma^t)^2)(z)\} \\ 0, & otherwise \end{cases}$$

M-step:

$$\mu_\mathbf{i} = (I - \xi_i V Q (I + \xi \Sigma)^{-1} Q' V')(\alpha_\mathbf{i} - \frac{\sum_j n_{ij} \nu_\mathbf{j}}{\sum_j n_{ij}})$$

$$\nu_{\mathbf{j}} = (I - \zeta_j MU(I + \zeta_j \Lambda)^{-1} U' M')(\beta_{\mathbf{j}} - \frac{\sum_i n_{ij}\mu_{\mathbf{i}}}{\sum_i n_{ij}})$$

3.3.2 CAMI

An information theoretic approach, called CAMI, based the concepts of *mutual information* (A.1) and *maximul likelihood* (A.1), is presented in [DB10a]. The approach, which produces simultaneously two different clusterings, is optimising an objective function which combines quality and dissimilarity, as other methods we've previously discussed do. Maximum likelihood is used to ensure quality, while mutual information ensures dissimilarity since it is minimised between the two different clustering solutions produced.

Any clustering solution is seen as a mixture of models, where each distribution in the mixture corresponds to a cluster, and the cluster label C is seen as the missing data Y in the EM algorithm (A.1). Given a dataset $X \in \mathbb{R}^{d \times n}$, the method produces two clustering solutions C_1 and C_2 parameterised by Θ_1 and Θ_2 respectively, which partition the set into two groups M_1 and M_2 whose similarity is minimised. Let Θ be a combination of Θ_1 and Θ_2, the log-likelihood function is:

$$\tilde{L}(\Theta; X) = L(\Theta_1; X) + L(\Theta_2; X) - \eta I(C_1; C_2|\Theta)$$

the log-likelihood terms $L(\Theta_1; X)$ and $L(\Theta_2; X)$ correspond to the quality of the two clusterings while the mutual information term $I(C_1; C_2|\Theta)$ to their dissimilarity. The parameter $\eta > 0$ balances the trade-off between dissimilarity and quality.

Assuming that the partitions are independent, the mutual information is the pairwise mutual information between the two clustering solutions becomes $I(C_1; C_2|\Theta) = \sum_{i,j} I(c_{1i}, c_{2j}|\theta_{ij})$ where c_{1i} is the i^{th} cluster from the first clustering and c_{2j} the j^{th}

cluster from the second clustering. The authors also define the mutual information of a cluster c_{1i} to the clustering C_2 as:

$$I(c_{1i}; C_2) = \sum_{c_{2j} \in C_2} p(c_{1i}, c_{2j}) \log \frac{p(c_{1i}, c_{2j})}{p(c_{1i}) p(c_{2j})}$$

and from the properties of mutual information the objective function becomes:

$$\tilde{L}(\Theta; X) = L(\Theta_1; X) + L(\Theta_2; X) - \eta \sum_{i,j} p(c_{1i}, c_{2j}) \log p(c_{1i}, c_{2j})$$

A variant of the EM algorithm (A.1) is then applied (the details of the algorithm are out of the scope of this dissertation and can be found in the original paper [DB10a]).

Chapter 4

Our approach

"The quiet statisticians have changed our world - not by discovering new facts or technical developments but by changing the ways we reason, experiment and form our opinions about it."

Ian Hacking

4.1 An overview of the data mining framework

Our approach in alternative clustering falls into the context of a general information theoretic data mining framework presented in [De 11a] by Tijl de Bie. It is crucial to briefly overview the main concepts of this framework in order for the reader to get an insight that is essential towards the understanding of our approach in the following sections.

In this framework, *data mining* is defined [De 11a] as *"a process of information exchange between the user and the data"*. The goal of data mining is provide the user with a good understanding of the data, or in other words, to reduce his uncertainty

about the data. Although there have been numerous approaches with a similar goal, the main difference of this framework is that it uses the subjective nature of *interestingness*. It doesn't try to objectively define what is interesting, but instead, it takes into account user's beliefs and tries to quantify interestingness with respect to the user. The more interesting a pattern that's revealed to the user is, the more it will reduce her uncertainty about the data. To do so, the framework proposes a way to quantify the concepts of *"interestingness"* and *"uncertainty"*.

Specifically, user's prior beliefs about the data $\mathbf{X} \in \mathbb{R}^{d \times n}$ are modelled as a distribution P which we will refer to as *the background distribution*. However the user is unable to formalise her beliefs into the distribution P directly. For that reason, user's beliefs are formalised as constraints that this distribution must satisfy. In this way, the domain of possible distributions is restricted to a set of distributions, those who satisfy these constraints. It's easy to see that the size of this set of possible distributions is inverse proportional to user's prior beliefs.

In our case, these constraints will be that certain statistics f_i (e.g. cumulants) are already known:

$$\int_{\mathbf{X} \in \mathbb{R}^{d \times n}} f_i(\mathbf{X}) P(\mathbf{X}) = c_i, \forall i \in \mathbb{N}, c_i \in \mathbb{R}$$

To choose the specific distribution, that will best express user's beliefs, from the set $\mathcal{P}$ of possible distributions, it's plausible to choose the most unbiased one, the maximum entropy (denoted as *MaxEnt*) distribution

$$P = \arg\max_{P \in \mathcal{P}} \quad -E_{X \sim P}\{\log(P(X))\}$$

since any other distribution from $\mathcal{P}$ will have a lower entropy than P from the *MaxEnt* distribution, which means that more knowledge is possible be added that will lead to further reduce of uncertainty.

Before we proceed with the formalisation used to quantify the *"interestingness"* of a pattern, we first need to define what a pattern is. In the context of this framework, we regard patterns as constraints. A constraint $\mathbf{X} \in \mathcal{X}'$ restricts the set of possible values for the data into a subset of the data space $\mathcal{X}' \subseteq \mathcal{X} \in \mathbb{R}^{d \times n}$ and consequently reduces the user's uncertainty about the data. Providing a pattern to the user, the background distribution P will be updated to a new distribution P'.

The framework defines a measure of "interestingness" for a pattern as the negative log probability that the pattern exists in the data i.e. $-\log(P(\mathbf{X} \in \mathcal{X})$. In english, that means that the smaller the probability of a pattern existing in the data is, the more interesting this pattern is.

In the following sections we will explain how we applied the concepts of this framework in alternative clustering.

4.2 Prior beliefs

In our approach we express user's prior beliefs as constraints on the first and second order cumulants of the data points. This means that the user has some knowledge about the means and the variance of certain data points. It has to be noted that this can either be a right or wrong belief about the data or some calculated values based on the data.

From the family of all distributions that satisfy these constraints $\mathcal{P}$ we choose the most unbiased one which, according to the *principle of maximum entropy*, is the maximum entropy distribution. This is a multivariate Gaussian distribution with the specified means vector and covariance matrix.

$$P(\mathbf{X}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi^k|\Sigma|}} \cdot \exp\left[-\frac{1}{2}(x_i - \mu)'\Sigma^{-1}(x_i - \mu)\right]$$
$$= \frac{1}{\sqrt{2\pi^{kn}|\Sigma|}} \cdot \exp\Big(-\frac{1}{2} \cdot trace[(\mathbf{X} - \mathbf{e}\mu')\mathbf{\Sigma^{-1}}(\mathbf{X} - \mathbf{e}\mu')']\Big)$$

4.3 The clustering pattern

According to the data mining framework [Bie00], that our approach is based, patterns are properties of the data. Thus, their presence reduces the uncertainty of the user about the data by restricting the domain of possible values the data may have.

For the scope of this dissertation we choose a specific type of *clustering pattern*, containing k clusters, as a constraint of the form $\mathbf{X'E} = \mathbf{M}$, where $\mathbf{X} \in \mathbb{R}^{d\times n}$ are the data points, $\mathbf{E} \in \{0,1\}^{n\times k}$ is an indicator matrix, that contains cluster indicators as columns, and $\mathbf{M} \in \mathbb{R}^{d\times k}$ which contains the means vectors $\mu_\mathbf{i}$ as columns.

$$\mathbf{X'} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix} = \left[\mu_\mathbf{1}, \mu_\mathbf{2}, \cdots, \mu_\mathbf{k}\right]$$

We will use the term "pattern" for the *clustering pattern* unless otherwise stated.

4.4 Self information of a clustering pattern

As we stated earlier, according to the data mining framework [Bie00], we define the self-information of a pattern as the negative log probability of the pattern exists in

the data. We present a theorem that helps us quantify the self-information of our clustering pattern (the proof can be found in [De 11b]):

Theorem 1. Let the columns of the matrix $\mathbf{E}$ be the indicator vectors of a set of k clusters $\mathcal{I} = I_i$, $i = 1, \cdots, k$, and $\mathbf{P_E} = \mathbf{E}(\mathbf{E}'\mathbf{E})^{-\mathbf{1}}\mathbf{E}'$, the projection matrix onto the column space of $\mathbf{E}$. Then, the probability of the pattern $\mathbf{X}'\mathbf{E} = \mathbf{M}$ is given by:

$$P(\mathbf{X}'\mathbf{E} = \mathbf{M}) = \frac{1}{\sqrt{(2\pi)^{kd}|\mathbf{\Sigma}^k}} \exp\big(-\frac{1}{2} trace[\mathbf{P_e} \cdot (\mathbf{X} - \mathbf{e}\mu')\mathbf{\Sigma}^{-\mathbf{1}}(\mathbf{X} - \mathbf{e}\mu')']\big)$$

Thus, the self-information of a pattern defined by the columns of $\mathbf{E}$, which is its negative log probability and denoted as $SelfInformation_{\mathcal{I}}$, is equal to:

$$SelfInformation_{\mathcal{I}} = \frac{k}{2}\log\big((2\pi)^d|\Sigma|\big) + \frac{1}{2}Q_{\mathcal{I}}$$

where

$$Q_{\mathcal{I}} = trace[\mathbf{P_e} \cdot (\mathbf{X} - \mathbf{e}\mu')\mathbf{\Sigma}^{-\mathbf{1}}(\mathbf{X} - \mathbf{e}\mu')']$$

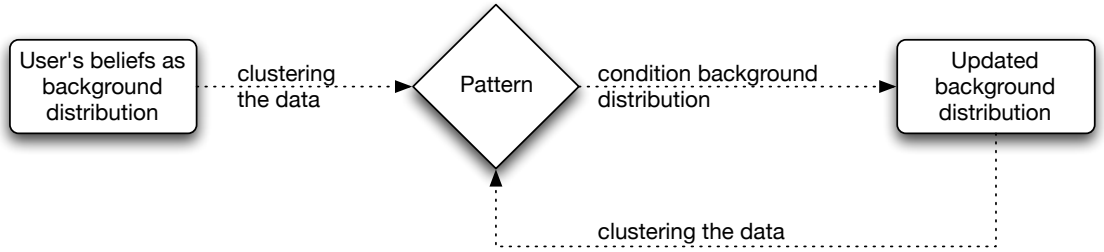
Since, the self-information depends on $\mathcal{I}$ only through $Q_{\mathcal{I}}$, we use $Q_{\mathcal{I}}$ as the **quality metric** for a pattern.

Based on this theorem, we can quantify the self-information of a pattern. However, searching for the most informative pattern over the set of all possible patterns, i.e. the pattern that maximises our quality metric, is an NP-hard optimisation problem.

Nonetheless, from an algorithmic viewpoint, this problem can be reduced in a *weighted set coverage problem* [De 11a] and thus, although it can't be solved directly, a proven good approximation can be achieved using an iterative greedy search algorithm [CLRS01], i.e. an *iterative data mining* approach.

4.5 The iterative data mining approach

FIGURE 4.1: A graphic representation of our approach.



In order to approximately solve the optimisation problem, we built an iterative search technique based on spectral clustering.

More specifically, in each iteration, we are searching for the most informative pattern (clustering) based on our quality metric. However, after the first iteration, where the first clustering is found, we need to find the second most informative pattern keeping the previous patterns as they are. In other words, in each iteration we are searching for the most informative pattern given the previously found patterns which in fact is a sequential alternative clustering method using side information.

Let $\mathbf{Q_E} = \mathbf{I} - \mathbf{P_e}$ be the projection matrix on the kernel of $\mathbf{E}$, then based on the definition of the projection matrix and the *matrix inversion lemma* [Woo50], each iteration reduces to the maximisation of the following increase of the quality matrix:

$$\Delta Q_{\mathcal{I}} = \frac{\mathbf{Q_E} \cdot (\mathbf{X} - \mathbf{e}\mu')'\mathbf{\Sigma^{-1}}(\mathbf{X} - \mathbf{e}\mu')' \cdot \mathbf{Q_E}}{\mathbf{Q_E}} \tag{4.5.1}$$

Thus, the objective we need to optimise is a Rayleigh quotient. Relaxing the requirement that the matrix $\mathbf{E}$ contains only binary values to real values, we reduce the problem of optimising this Rayleigh quotient into an eigenvalue problem. This means, that (4.5.1) is maximised by the dominant eigenvector of the matrix $\mathbf{Q_E} \cdot (\mathbf{X} - \mathbf{e}\mu')'\mathbf{\Sigma^{-1}}(\mathbf{X} - \mathbf{e}\mu')' \cdot \mathbf{Q_E}$.

The technique presented in [De 11b] uses an exhaustive to threshold the real values into binary values. However, in our case, this is impossible since we are using the matrix $\mathbf{E}$ instead of just a vector and performing an exhaustive search for even small data sets is computationally unfeasible.

For this reason, in each iteration, we will use as a base of our searching algorithm on a well known spectral clustering algorithm, presented in [NJW01]. For simplicity, we use prior beliefs of means equal to zero and covariance matrix equal to the identity matrix i.e. $\mathbf{\Sigma} = \mathbf{I}$. The user is able to change these assumptions based on her prior beliefs.

We first present the spectral algorithm that we use in each iteration of our alternative clustering setting. The difference from the original algorithm presented by Ng, Jordan & Weiss [NJW01] is in the similarity matrix we are using. Instead of using an affinity matrix and calculating the Laplacian matrix, we use directly our quality metric matrix as the affinity matrix.

Algorithm 3 Spectral clustering

Given a quality metric matrix $\mathbf{C}$, cluster data $\mathbf{X} = \{x_1, \cdots, x_n\}, x_i \in \mathbb{R}^d$ into k clusters.
1. Find the k dominant eigenvectors of $\mathbf{C}$ and form the matrix $K = [x_1 x_2 \cdots x_k]$ which contains the k dominant eigenvectors as columns.
2. Form the matrix $\mathbf{U}$ by normalising the rows of $\mathbf{K}$ to have unit length (i.e. $U_{ij} = K_{ij}/(\sum_j K_{ij}^2)^{\frac{1}{2}}$)
3. Cluster the rows of $\mathbf{U}$ in k clusters using K-means algorithm
4. Assign each point x_i to cluster j if and only if row i of the matrix $\mathbf{U}$ was assigned to cluster j

We present the algorithm of our approach. Please note that in the first iteration, $\mathbf{Q_E} = I$ by the definition of $\mathbf{Q_E}$ which means that the matrix we need to optimise is $\mathbf{XX}'$.

Algorithm 4 Alternative clustering

Given the data set $\mathbf{X} \in \mathbb{R}^{d \times n}$, present alternative clusterings (until user is satisfied), each of which contains k clusters

1. Calculate the quality metric matrix $\mathbf{C} = \mathbf{X}\mathbf{X}'$
2. Perform spectral clustering (Algorithm 3)
3. Construct the indicator matrix E

loop

Calculate the quality metric matrix $\mathbf{C} = \mathbf{Q_E} \cdot \mathbf{X}\mathbf{X}' \cdot \mathbf{Q_E}$

Perform spectral clustering (Algorithm 3)

Stack the new indicator matrix $\mathbf{E}$ to the previous matrix $\mathbf{E}$

end loop

4.5.1 Kernel based version

Under the assumption (that we also did for the scope of this dissertation) that the covariance matrix of the multivariate Gaussian distribution equals the identity matrix i.e. $\mathbf{\Sigma} = \mathbf{I}$, the quality metric $\mathbf{Q}_{\mathcal{I}}$ depends on the data set $\mathbf{X}$ only through the inner product $\mathbf{X}\mathbf{X}'$. This allows us to derive a kernel version of our approach just by replacing this inner product with a suitable kernel matrix. In fact, we used the radian basis function (RBF) as kernel in order to enable our method to obtain non-linearly shaped clusters.

Chapter 5

Validation and results

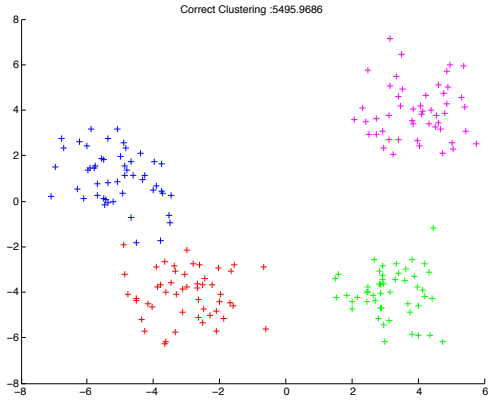
"Computers are useless. They can only give you answers."

Pablo Picasso

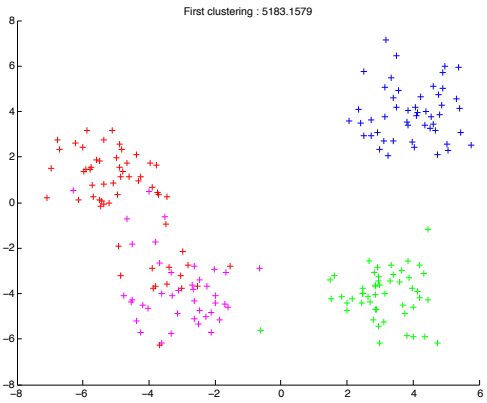
5.1 Validation

In order to validate that our quality metric holds, we produce some synthetic data, knowing the correct clustering (ground truth). We calculate the value of our quality metric for the correct clustering and we compare it against the value of the quality metric of the first clustering produced (the one with the biggest quality metric) by our method. We perform this experiment 1,000 times and we take the average to exclude any results by chance.

The synthetic dataset we produced contains 200 data points in 4 different clusters, sampled from 4 Gaussian distributions (one for each cluster) having identity covariance matrix and different means.



(a) One correct clustering



(b) The first clustering produced by our method corresponding to the correct clustering above

The results over 1,000 repetitions (Figure 5.1) in:

- Quality metric of the *correct* correct clustering: $\mathbf{5.4404} \times 10^3$
- Quality metric of the *first* produced clustering: $\mathbf{5.1228} \times 10^3$
- Quality metric of a *random* produced clustering: $\mathbf{212.5209}$

Thus, we can conclude that our quality metric makes sense. For a discovered clustering, the quality metric is close to the value of the correct clustering. This is emphasised when it is compared to the value of the quality metric for a random clustering.

We did expect the value for the quality metric for the discovered clustering to be slightly lower than the value for the correct clustering since we use an approximate and not an exact method.

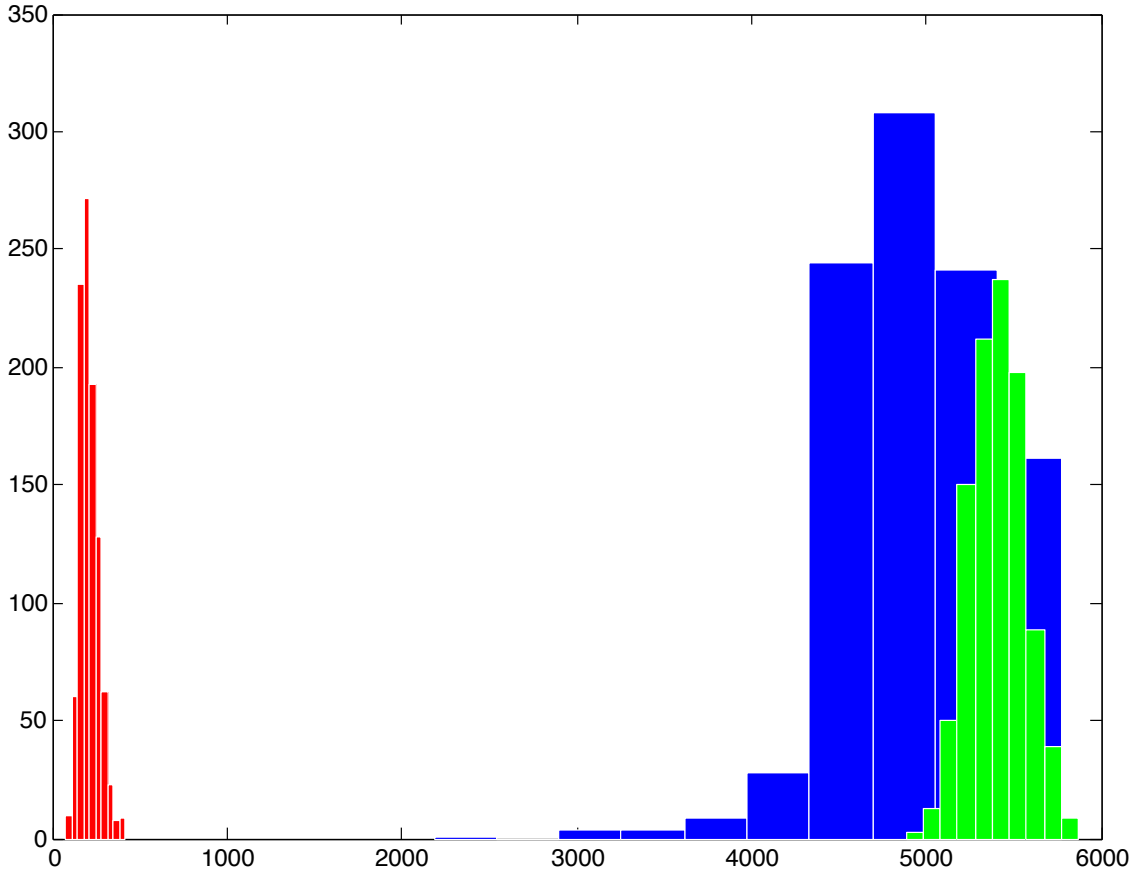


FIGURE 5.1: Results of 1,000 repetitions. On axis x is shown the quality metric and on axis y the frequency. Bars in red represent the value of the quality metric for the random clustering, in green for the correct clustering and in blue for the first clustering found by our approach.

5.2 Synthetic data with 4 clusters

We create a synthetic dataset of 200 data points, containing 4 clusters, each of which is sampled from a Gaussian distribution with different means and identity covariance matrix. We choose to cluster the data into two clusters since that way, alternative clusterings will more meaningful.

We perform two experiments. In the first, we are using the inner product $\mathbf{X}\mathbf{X}'$ as

described by our quality metric, whilst in the second, we replace it with an RBF kernel with kernel width of 3. In the figures, above the clustering, we present the value of the quality metric for the corresponding clustering.

5.2.1 Using inner product

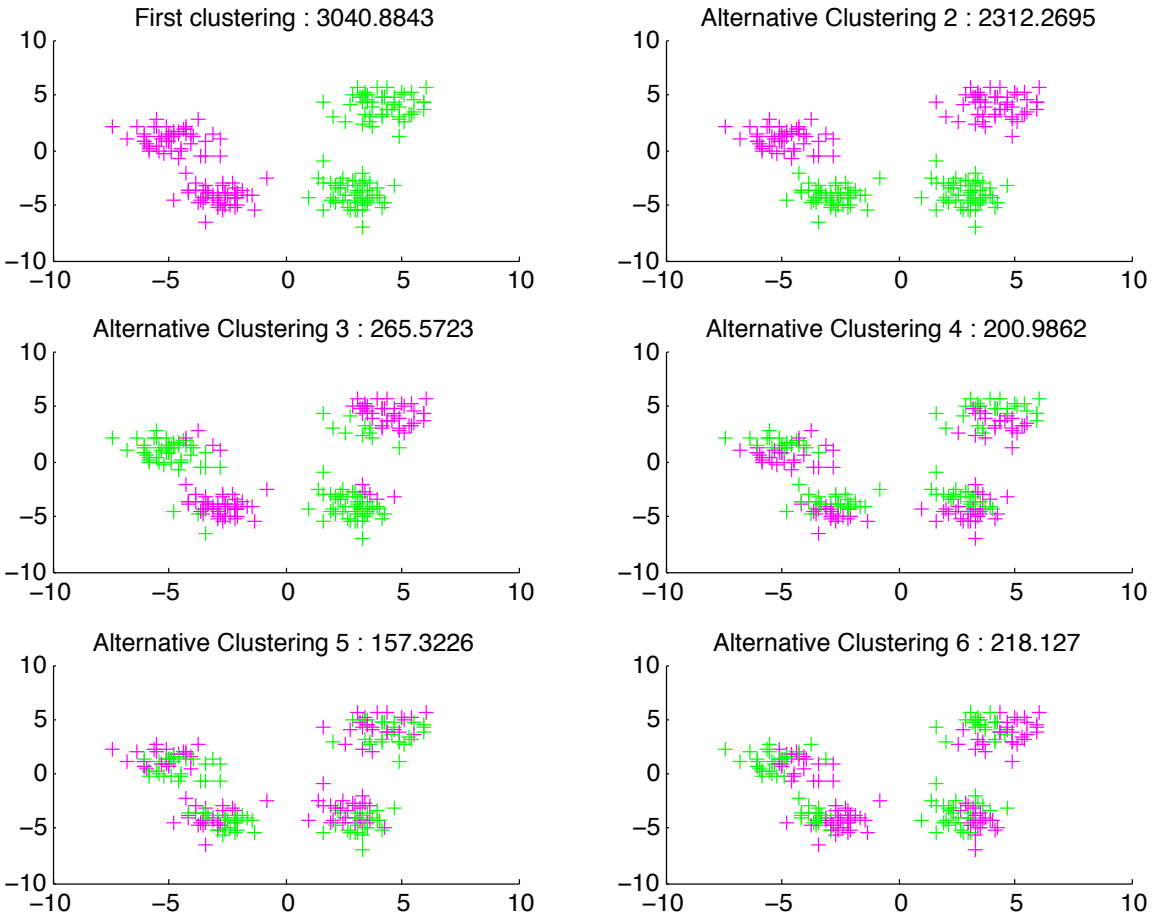


FIGURE 5.2: Using quality metric with inner product **XX'**

We see that the first two clusterings produced are of high quality and how the quality metric decreases for the next clusterings which are of lower quality.

5.2.2 Using RBF kernel

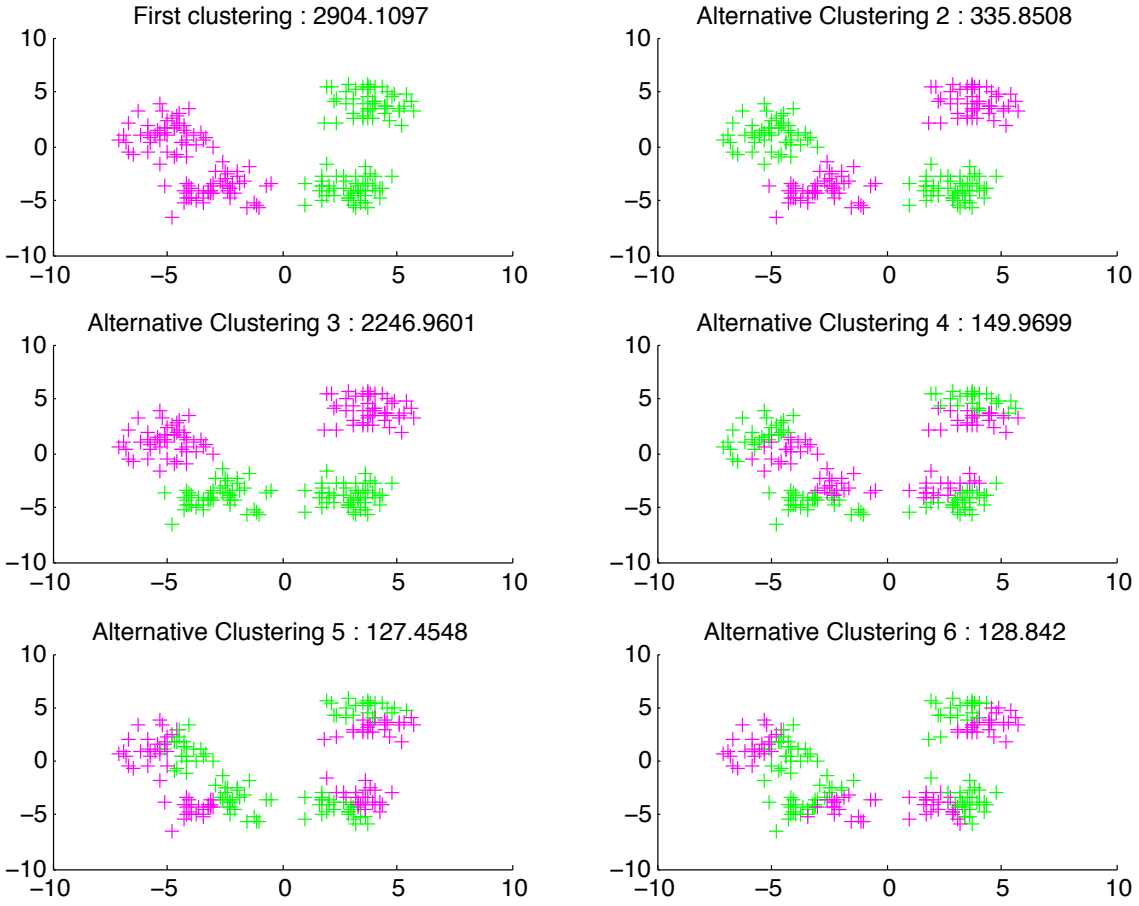


FIGURE 5.3: Using quality metric replacing the inner product **XX'** with an RBF kernel of width 3

Similarly to the previous experiment, the method finds two high quality alternative clusterings.

5.3 Synthetic data with 3 clusters

We create a synthetic dataset of 200 data points, containing 3 clusters, each of which is sampled from a Gaussian distribution with different means and identity covariance matrix. We choose to cluster the data into two clusters since that way, alternative clustering will more meaningful.

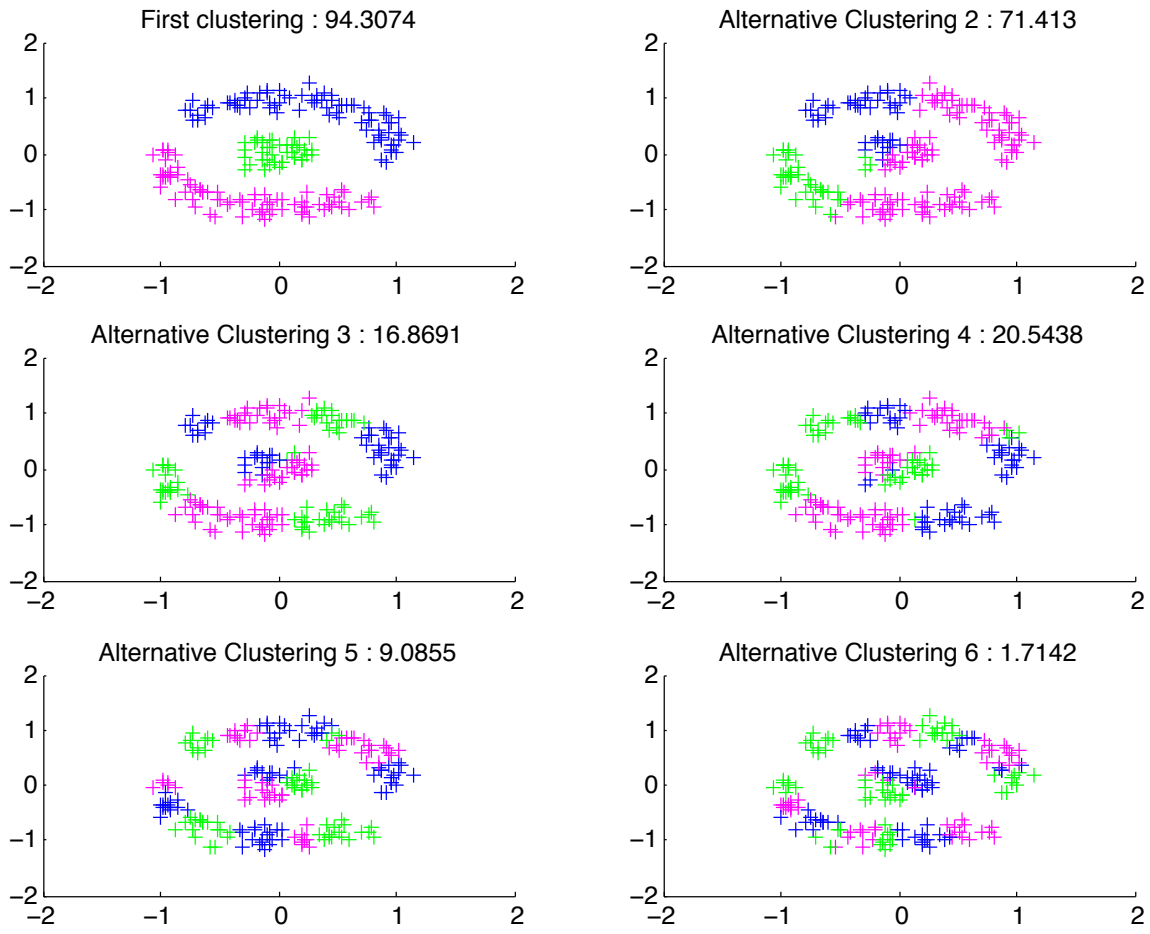


FIGURE 5.4: Using RBF kernel in a 3 clusters dataset

In this experiment, the RBF kernel is used in order to get non linear clusterings. As previously, two high quality views of the data can be found before the quality metric drops to a much lower level.

Chapter 6

Conclusion and future directions

"And if you find her poor, Ithaca has not deceived you.
Wise as you have become, with so much experience,
you must already have understood what Ithacas mean."

Constantine P. Cavafy

6.1 Conclusion

The first aim of this project was to provide an overview of the most important alternative clustering approaches. We concluded to a categorisation of these methods to three main categories: *semi-supervised*, *data-transformation oriented* and *unsupervised*. We discussed the main points of each method trying to touch on what each methods does differently. Ideally, we could have co,pared all the methods by using a single data set and provide the results. This was not done for two main reasons. First, developing all these methods from scratch was out of the time interval given for this dissertation and secondly and most importantly, we believe that this would not be

the most objective way to compare them since each method has different performance for different kind of dataset (e.g. sparse data, high-dimensional data etc).

The second aim of this project was to define a new approach, extending a previous method in a certain framework[De 11a]. Our approach differs from that presented in [De 11b], in two major points. Firstly, we used a different pattern syntax, we defined a *clustering* pattern and secondly we presented a new approximation search algorithm based on spectral clustering. We defined our approach theoretically, explaining its different aspects and crucial points. Also, we conducted experiments on synthetic data, arguing about the validation of the quality criterion that characterised our whole approach, and we succeeded in finding at least one alternative clustering of high quality on our experiments.

6.2 Future directions

Alternative clustering is a new field, that is now starting to get attention in the literature. That means that many more alternative clustering techniques will be produced and soon this survey will be outdated. Furthermore, it would be of great interest to define an objective way to compare these methods in an as objective as possible way.

The method we proposed in this dissertation is an approximate method. That means, they is subject to improvements. This may achieved using tighter relaxations (e.g. Semi Definite Programming (SDP), 0-1 SDP etc). Another extension would be the use of other forms of prior beliefs, not just constraints a probability distribution must satisfy. As a consequence, the syntax of the pattern used, could be altered. A third extension, initially as described in [De 11b]; a cost, i.e. description length, of a clustering pattern could also be taken into account, while we could explore the use of different costs that could be appropriate for different patterns. Finally, more

experiments, ideally using real world data could be performed to extract more useful insight of the method as well as have some real world results.

Appendix A

Mathematical preliminaries

A.1 Information theoretic concepts

Multivariate Gaussian distribution

The *multivariate Gaussian distribution* is a generalisation of the Gaussian distribution for more than one variables. A random vector $\mathbf{X} = [X_1, X_2, \cdots, X_n]'$ is said to have multivariate Gaussian distribution ($\mathbf{X} \backsim \mathcal{N}(\mu, \Sigma)$) with mean $\mu \in \mathbb{R}^n$ and covariance matrix $\mathbf{\Sigma} \in S^n_{++}$ [1] if its probability density function (pdf) is given by

$$p(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\Big(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\Big)$$

Vital concept towards the understanding of the multivariate Gaussian distribution, is that of the *covariance matrix*. The covariance matrix $\mathbf{\Sigma}$ of a random vector $\mathbf{X}$ contains in its i^{th} and j^{th} position the covariance between the i^{th} and j^{th} random variable.

[1] $S^n_{++} = \{A \in \mathbb{R}^{n \times n} : A = A' \text{ and } x'Ax > 0 \forall x \in \mathbb{R}^n \text{ such that } x \neq 0 \}$

$$\Sigma = Cov(X_i, X_j) = E[(X_i - E(X_i)) \cdot (X_j - E(X_j))] = E[X_i X_j] - E[X_i]E[X_j]$$

Entropy and conditional entropy

In information theory, entropy is a measure of uncertainty of a random variable. More formally, let X be a discrete random variable (or a continuous random variable use the integral instead of the sum) with probability mass function $p(x) = P(X = x), \ \ x \in X$. The entropy $H(X)$ is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) = E_p\{-\log(p(X))\}$$

we can also denote the entropy $H(X)$ as $H(p)$.

The remaining uncertainty of a random variable Y when X is given, where $(X, Y) \sim p(x, y)$, is the *conditional entropy*:

$$H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{x \in \mathcal{Y}} p(x, y) \log p(y|x) = -E\{\log(p(Y|X))\}$$

Maximum entropy distribution

The *maximum entropy distribution (MaxEnt)* is the distribution which has entropy at least as great as all the other members amongst a specified family of distributions. We are looking the probability density p that maximises the entropy $H(p)$.

As described in [Bie10], we can formalise this process as an optimisation problem:

$$\begin{aligned} \max_{p(x)} \quad & -\sum_{x \in X} p(x) \log p(x) \\ s.t. \quad & -\sum_{x \in X} f_i(x) = a_i, \forall i \\ & -\sum_{x \in X} p(x) = 1 \end{aligned}$$

Kullback-Leibler divergence

The measure of distance between two distributions with probability mass functions $p(x)$ and $q(x)$, known as *Kullback-Leibler divergence* is defined as:

$$KL(p||q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = E_p\{\log(p(x)) - \log(q(x))\}$$

Mutual information

The *mutual information* between two random variables X, Y with a joint probability distribution $p(x, y)$ and marginal probability mass functions $p(x)$, $p(y)$ is the Kullback-Leibler distance between the joint distribution and the product distrubution $p(x)p(x)$

$$I(X;Y) = KL\big(p(x,y)||p(x)p(y)\big) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

Maximum likelihood estimation

Maximum likelihood estimation (MSE) is a technique used to estimate the parameters of a statistical model. The goal is to find Θ that maximises the log-likelihood function (also referred to as the cross-entropy):

$$L(\Theta; X) = \sum_{i=1}^{n} \log p(x_i|\Theta)$$

where $X \in \mathbb{R}^{d \times n}$ the iid [2] observations from the distribution $p(x|\Theta)$.

EM algorithm

The EM algorithm is a technique for iteratively computing the MSE when data X is incomplete and there exists another dataset Y corresponding to the missing (and unknown) data. The technique has two steps, the *E-step* (expectation step) and the *M-step* (maximisation step) and tries to maximise the likelihood of the combined (X and Y) data.

Specifically, the E-step:

$$Q(\Theta|\Theta_t) = E[\log p(X, Y|X, \Theta_t$$

where the algorithm determines the expectation of the log-likelihood based on the current parameter Θ_t and the M-step:

$$\Theta_{t+1} = \arg\max_{\Theta} Q(\Theta|\Theta_t)$$

where the algorithm finds a new parameter Θ_{t+1} that maximises this quantity.

[2] Independent and identically distributed

A.2 Linear algebra concepts

Rayleigh quotient

Definition 2. The *Rayleigh quotient* to the *Hermitian* matrix **A** is defined [HJ94] as

$$R_A(x) = \frac{x^* Ax}{x^* x}$$

where x^* is the Hermitian conjugate of x.

The importance of this quantity lies in the fact that since **A** is a Hermitian matrix, it has real eigenvalues and for a given matrix **A** the Rayleigh quotient is maximised by the dominant eigenvector.

Matrix inversion lemma

The matrix inversion lemma (also known as Woodbury matrix identity) is

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U^{-1})VA^{-1}$$

for matrices A, U, C and V of the correct size. A proof can be found in [BV04].

Appendix B

Parts of the source code

Here we present some parts of the Matlab code used for the experiments. The RBF kernel code is written by Tijl De Bie and can be found in http://www.kernel-methods.net/matlab/kernels/rbf.m.

B.1 Matlab code for synthetic data generation

```
function [data] = GenerateData(select,n)

if select == 1
    K=n;
data=[randn(n/4,1)+8 randn(n/4,1)+12 ;
   randn(n/4,1)+3 randn(n/4,1)-4 ;
   randn(n/4,1)-15 randn(n/4,1)+11 ;
   randn(n/4,1)-10 randn(n/4,1)-10 ];

elseif select == 2
    data=zeros(n,2);
    r=linspace(0,1,n/5);
    data(1:n/5,1)=cos(r*2*pi)*0.2;
    data(1:n/5,2)=sin(r*2*pi)*0.2;
    r=linspace(0,1,2*n/5);
    data(n/5+1:3*n/5,1)=cos(r*pi*0.75)*1;
    data(n/5+1:3*n/5,2)=sin(r*pi*0.75)*1;
    r=linspace(0,1,2*n/5);
    data(3*n/5+1:n,1)=cos(pi+r*pi*0.75)*1;
    data(3*n/5+1:n,2)=sin(pi+r*pi*0.75)*1;
    data=data+randn(n,2)*0.1;
elseif select == 3
    data=[randn(n/4,1)+4 randn(n/4,1)+4 ;
    randn(n/4,1)+3 randn(n/4,1)-4 ;
    randn(n/4,1)-5 randn(n/4,1)+1 ;
    randn(n/4,1)-3 randn(n/4,1)-4 ];


end
```

B.2 Matlab code for an experiment

```
clear all;
close all;

n=200;
data = GenerateData(2,n);

%C=data*data';
C=rbf(data,0.2);

% Number of clusters
k=2;

%% Calculate quality for the
%correct clustering
%(only for 4 clusters data)

% Number of clusters (in correct)
kk = 4;

Ecor = zeros(n, kk);

Test = [ones(n/4,1); ones(n/4,1)*2;
ones(n/4,1)*3; ones(n/4,1)*4];

for j=1:kk
    for i=1:n
        if Test(i,1) == j
            Ecor(i,j) = 1;
        end
    end
end

Qcor = trace(Ecor*pinv(Ecor)*(data*data'));

%figure,scatter(data(:,1),data(:,2),'o')

figure,
hold on;
for i=1:n
```

```
    if Ecor(i,1) == 1
        plot(data(i,1),data(i,2),'m+');
    elseif Ecor(i,2) == 1
        plot(data(i,1),data(i,2),'g+');
    elseif Ecor(i,3) == 1
        plot(data(i,1),data(i,2),'b+');
    else
        plot(data(i,1),data(i,2),'r+');
    end
end
hold off;
title(['Correct Clustering :', num2str(Qcor)]);

%% First clustering

% Find the k dominant eigenvectors

[v,d] = eigs(C,k,'LM');

% Create normalised matrix U using
%the eigenvectors found

for i=1:size(v,1)
    t = sqrt(sum(v(i,:).^2));
    U(i,:) = v(i,:) ./ t;
end

% Perform k-means clustering

[IDdata,Z] = kmeans(U,k);

% Construct matrix E

E = zeros(n,k);


for j=1:k
    for i=1:n
        if IDdata(i,1) == j
            E(i,j) = 1;
        end
    end
```

```
end

Q = trace(E*pinv(E)*(data*data'));

% An index (not binary) for the
%clustering

IDX = IDdata;

% Plot

f=figure;
subplot(3,2,1)
hold on;
for i=1:n
    if E(i,1) == 1
        plot(data(i,1),data(i,2),'m+');
    elseif E(i,2) == 1
        plot(data(i,1),data(i,2),'g+');
    elseif E(i,3) == 1
        plot(data(i,1),data(i,2),'b+');
    else
        plot(data(i,1),data(i,2),'r+');
    end
end
hold off;
title(['First clustering : ', num2str(Q)]);

%% Subsequent clusterings
for z = 2:6;
    C = (eye(n)-E*pinv(E))*C*(eye(n)-E*pinv(E));

    % Find the k dominant eigenvectors

    [v,d] = eigs(C,k,'LM');

    % Create normalised matrix U using
    %the eigenvectors found

    for i=1:size(v,1)
        t = sqrt(sum(v(i,:).^2));
        U(i,:) = v(i,:) ./ t;
```

```
    end

    % Perform k-means clustering

    [IDdata,Z] = kmeans(U,k);

    % Add k new columns at E in each iteration

    Efoo = zeros(n,k);

    for j=1:k
        for i=1:n
          if IDdata(i,1) == j
            Efoo(i,j) = 1;
          end
        end
    end

    E = [E Efoo];

    Qfoo = trace(Efoo*pinv(Efoo)*(data*data'));

    Q = [Q Qfoo];

    IDX = [IDX IDdata];

    % Plot

    figure(f)
    subplot(3,2,z)

    hold on;
    for i=1:n
        if Efoo(i,1) == 1
            plot(data(i,1),data(i,2),'m+');
        elseif Efoo(i,2) == 1
            plot(data(i,1),data(i,2),'g+');
        elseif Efoo(i,3) == 1
            plot(data(i,1),data(i,2),'b+');
        else
            plot(data(i,1),data(i,2),'r+');
        end
```

```
166     end
167     hold off;
168     title(['Alternative Clustering ',
169     num2str(z),' : ',
170     num2str(Qfoo)]);
171
172 end
```

Bibliography

[BB06] Eric Bae and James Bailey. COALA: A Novel Approach for the Extraction of an Alternate Clustering of High Quality and High Dissimilarity. *Sixth International Conference on Data Mining (ICDM'06)*, pages 53–62, December 2006.

[Bie00] Tijl D E Bie. An information theoretic framework for data mining. pages 1–15, 2000.

[Bie10] Tijl De Bie. Maximum entropy models and subjective interestingness : an application to tiles in binary databases. pages 1–43, 2010.

[Bis95] C.M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[Bs06] C.M. Bishop and SpringerLink (Online service). *Pattern recognition and machine learning*, volume 4. Springer New York, 2006.

[BV04] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004.

[CFD07] Y. Cui, X.Z. Fern, and J.G. Dy. Non-redundant multi-view clustering via orthogonalization. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, volume 3, pages 133–142. IEEE, 2007.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. 2001.

[DB10a] X.H. Dang and James Bailey. Generation of alternative clusterings using the CAMI approach. In *SIAM International Conference on Data Mining (SDM)*, pages 118–129, 2010.

[DB10b] Xuan Hong Dang and James Bailey. A Hierarchical Information Theoretic Technique for the Discovery of Non Linear Alternative Clusterings. *Optimization*, 2010.

[De 11a] T. De Bie. An Information Theoretic Framework for Data Mining. *Communications*, 2011.

[De 11b] T. De Bie. Subjectively interesting alternative clusters. Technical report, Technical report, University of Bristol TR-133090, 2011.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[DQ08] Ian Davidson and Zijie Qi. Finding Alternative Clusterings Using Constraints. *2008 Eighth IEEE International Conference on Data Mining*, pages 773–778, December 2008.

[Dun73] J.C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Cybernetics and Systems*, 3(3):32–57, 1973.

[FPSS+96] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, et al. Knowledge discovery and data mining: Towards a unifying framework. In *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR*, pages 82–88, 1996.

[GH03] David Gondek and Thomas Hofmann. Conditional information bottleneck clustering. In *3rd ieee international conference on data mining, workshop on clustering large data sets*, number C, pages 36–42. Citeseer, 2003.

[GH04] D. Gondek and T. Hofmann. Non-redundant data clustering. *Knowledge and Information Systems*, 12(1):1–24, 2004.

[GH05] David Gondek and Thomas Hofmann. Non-redundant clustering with conditional ensembles. *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining - KDD '05*, 1(c):70, 2005.

[HJ94] R.A. Horn and C.R. Johnson. *Topics in matrix analysis*. Cambridge Univ Pr, 1994.

[JD88] A.K. Jain and R.C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[JMD08] Prateek Jain, Raghu Meka, and Inderjit S Dhillon. Simultaneous Unsupervised Learning of Disparate Clusterings. *Analysis*, (April), 2008.

[JMF99] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, September 1999.

[Kle03] Jon Kleinberg. An impossibility theorem for clustering. In *Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference*, page 463. The MIT Press, 2003.

[NJW01] A. Ng, M. Jordan, and Y. Weiss. Spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems 14: Proceeding of the 2001 Conference*, pages 849–856, 2001.

[PM06] Anne Patrikainen and Marina Meila. Comparing subspace clusterings. *IEEE Transactions on Knowledge and Data Engineering*, 18(7):902–916, 2006.

[QD09] ZiJie Qi and Ian Davidson. A principled and flexible framework for finding alternative clusterings. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, page 717, 2009.

[TK08] S. Theodoridis and K. Koutroumbas. Pattern recognition. 2008.

[TPB00] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. *Neural Computation*, pages 1–16, April 2000.

[Voo86] E.M. Voorhees. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing & Management*, 22(6):465–476, 1986.

[Woo50] M.A. Woodbury. Inverting modified matrices. *Memorandum report*, 42:106, 1950.

[XNJR03] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, pages 521–528, 2003.

[XW05] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 16(3):645–78, May 2005.