Un-Supervised Estimation of Cluster Validity - Methods and Applications

Erel Levine
M.Sc. Thesis submitted to the Feinberg Graduate School
Weizmann Institute of Science

Research conducted under the supervision of **Prof. Eytan Domany**

November 1999

Acknowledgments

I would like to thank Professor Eytan Domany for an intriguing year. His devoted guidance and interesting suggestions had taught me a lot. I also wish to thank Gaddy Getz and Noam Shental for useful discussions and ideas. I really enjoyed working with them.

To my family and friends I am indebted for their support and encouragement.

Contents

Chapter 1

Introduction

One of the fundamental modes of understanding and interpreting data, for which no a priori knowledge of the underlying distribution is available, is that of organization into sensible groupings. Clustering analysis, the study of group structure with no prior knowledge, is therefore a major technique in exploratory data analysis.

Throughout the years many clustering algorithms have been proposed. Algorithms are based on graph theory, statistical pattern recognition, self-organization methods and more. In recent years several algorithms which are rooted in statistical mechanics have been introduced.

Application of different clustering algorithms to a certain data set often leads to different results. In part, this is the result of different assumptions the algorithms make about the structure of the data. The sensitivity of different algorithms to noise, which inherently exists in the data, is also a major contribution to the difference between their results.

The question of structure in the data is clearly a matter of resolution. Viewing the data from a distance, it may seem to be a single bunch of points. A flea taking a walk inside the same data set, on the other hand, may consider each point to form its own cluster. Usually, the interesting structure (if any) lies in mid-range resolutions.

Many clustering algorithms provide as their output some clustering structure, sometimes in the form of a hierarchy. Other algorithms possess the ability to deny the existence of any structure in the data. The user of most clustering algorithms, in any case, is not provided with a measure for the reliability of the proposed solution.

Partition into clusters should reflect the underlying structure of the data. In less fortunate cases, however, it may be only the result of the algorithm's tendency to cluster anything. In some cases, some structure exists in the data by chance : a slight modification of the data would wipe this structure out. In such a case, the identified structure may be due only to noise.

If the results of cluster analysis are to be taken seriously, one must be able to distinguish between these cases. For example, if our clustering analysis of DNA microarray data identifies a cluster of genes that contribute to the evolution of cancer, we may very well wish to be assured that this cluster indeed exists in the data before publishing this finding.

The concept of cluster validation refers to methods and indicators which attempt to evaluate the results of a cluster analysis. This evaluation should be quantitative and objective, if it is to have any meaning. The most common methods involve the definition of some validity index for each cluster, which is supposed to represent its reliability [1, 2, 3, 4]. Appendix A provides a brief look into the subject of validity indices, and suggests an index particularly suitable for *SPC* .

The definition of such an index often encapsulates some assumptions made about the data, and about what clusters should look like. The most commonly made assumption about clusters is that of compactness.

In this work we take a different approach, which is based on resampling techniques [5, 6]. We suggest a resampling scheme and a figure of merit, which should reflect the stability of clusters and clustering solutions to resampling. In a sense, this is an introduction of noise into the problem, without the need to postulate a noise model. Clusters which survive the presence of such a noise are believed to a genuine feature of the data.

Resampling methods have been introduces in the past for cluster validity in the context of fuzzy clustering [3]. In that case, resampling is used to determine the number of expected clusters, which is an external parameter of the algorithm. The leave-one-out scheme [7], and the similar *Jacknife* method [6], were also used in the context of clustering [8]. Another resampling scheme, known as the *Bootstrap method*, was introduced to clustering validation by Jain *et al.* [9]. However, in all those cases the application of resampling has been problem-specific. In contrast, our use of resampling is independent of both the problem and the clustering algorithm.

A large data set often consists of many clusters. Some of these clusters may be just the result of fluctuations in the data, while other clusters, of roughly the same size, may be highly correlated significant clusters. We propose to use resampling methods to distinguish between these kinds of clusters.

The scenario just presented is often the case when we try to cluster gene expression profiles, obtained using a novel technique of DNA microarrays (or DNA chips). We tested our methods on DNA microarray experimental data.

This work is organized as follows. In chapter 2 we give an introduction to the subject of clustering, and describe clustering algorithms utilized in our work. Chapter 3 describes the concept of cluster validity, and provides a simple example to demonstrate the need for such a process. In chapters 4 and 5 we describe our resampling scheme, and demonstrate its use in determining the optimal resolution to be used, as well as the optimal value of an external parameter of the clustering algorithm. Finally, in chapter 6 we test our validation methods on DNA microarray data sets.

Chapter 2

Clustering and SPC

Clustering analysis is the study of group structure with no prior knowledge. It thus falls into the category of *unsupervised learning*. It has applications in vast area of fields, including image classification [10], classification of proteins [11], and DNA microarrays data analysis.

The clustering problem is to find a partition of a data set into groups, such that each group indicates the presence of a distinct category in the data. The problem is formally stated as follows.

> Determine the partition of N given data-points $\{x_i\}_{i=1}^N$ into groups, called *clusters*, such that the data-points of a cluster are more similar, in some sense, to each other than to data-points in different clusters.

Data are provided to the clustering algorithms either as points in a D-dimensional metric space; or by a $N \times N$ similarity (or dissimilarity) matrix, where the ij element gives the similarity (dissimilarity) between the data points i and j.

The two main approaches to partitional clustering are called *parametric* and *non-parametric*. In the former some knowledge of the clusters' structure is assumed, and in most cases the ability to represent the data-points in a D-dimensional metric space is essential.

Sometimes assumptions about the structure of the data are incorporated in a *global criterion*. The goal of such algorithms is to find a clustering assignment which optimizes this criterion. These methods are concerned with two major questions: the definition of the criterion to optimize, and the method to optimize it.

In many cases of interest, however, there is no a priori knowledge about the data structure. Non-parametric methods are suitable to deal with such cases, as the criteria they employ are *local.* Agglomerative methods [1], described in section 2.2, are good examples for this kind of techniques. These algorithms, however, suffer from a variety of problems, among which the sensitivity to noise in the data and the lack of criteria to identify significant partitions are the most notable.

Many algorithms tend to create clusters even when no natural clusters exist in the data. The problem of *clustering-tendency* is to decide whether the clustering solution is "natural" in the data, or has been imposed by the clustering algorithm. This problem emerges both in parametric methods, where the number of clusters is assumed, and in agglomerative methods.

Recently, Blatt *et al.* [12, 13, 14] have introduced a new approach to clustering, based on the physical properties of a magnetic system. The method, described in section 2.3, is called *SPC* . This is a non-parametric method, which makes no assumption about the data, and provides a hierarchical organization of it. The number of "macroscopic" clusters at any level of the hierarchy is an *output* of the algorithm. In this thesis we used the *SPC* algorithm as our major clustering method.

2.1 Hierarchical clustering methods

A hierarchical classification is a nested sequence of partitions [1]. Any level of the sequence can be regarded as a refinement of it predecessor. In other words, different levels of the sequence represent classifications at different resolutions.

It is easiest to represent the results of such clustering methods by using a *dendrogram* (see, for example, figure 4.6). A dendrogram is a tree structure, consisting of layers of nodes. Each node represents a cluster, and the edges connect clusters which are nested into one another. Distances along the vertical axis are determined either by a resolution parameter (if available), or by the steps of the algorithm (*e.g.* the number of merged clusters in agglomerative methods - see below). A horizontal cut of the dendrogram creates a clustering at a certain resolution level.

Most methods which produce such a hierarchical structure provide no guidance about the resolution levels which are natural for the problem at hand. One can have as many clustering solutions as layers in the dendrogram. As one of its major features, the *SPC* algorithm provides an inherent method to overcome this difficulty, as we describe in section 2.3.

2.2 Agglomerative clustering methods

Agglomerative algorithms are methods to perform hierarchical clustering [1]. These algorithms start with fully disjoint clusters, placing each of the N points into a different cluster. The hierarchy is formed by successively merging clusters. Agglomerative clustering methods are widely used due to their conceptual and implementational simplicity.

The various agglomerative clustering methods can be described using the following scheme, known as Johnson's algorithm for hierarchical clustering [15];

Step 1. Assign each point to a different cluster $\omega_i = \{x_i\}$ and use the similarity between the points as the initial similarity matrix between the clusters $S(\omega_i, \omega_j) = S_{ij}$.

Step 2. Find the most similar pair of distinct clusters, say ω_μ and ω_ν.

Step 3. Merge clusters ω_μ and ω_ν, and call it ω_γ.

Step 4. Update the similarity matrix by deleting the rows and columns corresponding to clusters ω_μ and ω_ν, and calculate the similarities (see below) between the new cluster ω_γ and the other clusters.

Step 5. Stop if only one cluster, that contains all the points, is left; otherwise go to **Step 2**.

Since the agglomerative algorithms merge two clusters at each iteration, the resulting dendrogram is a binary tree. The number of iterations is always $n - 1$.

Various agglomerative clustering methods differ by the way they calculate the new similarities between the joined cluster and the other clusters in **Step 4**. For example, the widely-used *Average Linkage* (AVL) algorithm defines this similarity as the average of the pairwise similarity between all pairs of points in the two clusters,

$$S(\omega_\mu, \omega_\nu) = \frac{1}{n_\mu n_\nu} \sum_{\substack{x_i \in \omega_\mu \\ x_j \in \omega_\nu}} S_{ij} \quad , \tag{2.1}$$

where n_μ is the number of data-points in cluster ω_μ. The AVL algorithm belongs to a sub-family of agglomerative clustering algorithms called *Sequential Agglomerative Hierarchical Non-overlapping* clustering methods (SAHN). In these methods, the similarity between clusters can be computed using the similarities of the previous iteration of the algorithm. For example, the AVL similarity measure (Equation 2.1) can be calculated by

$$S(\omega_\mu \cup \omega_\nu, \omega_\gamma) = \frac{n_\mu}{n_\mu + n_\nu} S(\omega_\mu, \omega_\gamma) + \frac{n_\nu}{n_\mu + n_\nu} S(\omega_\nu, \omega_\gamma) \quad . \tag{2.2}$$

Calculating the new similarities using the previous ones make the SAHN algorithms relatively efficient.

Another member of the SAHN family is Ward's method, also called *the minimum variance method.* This method defines the the similarity between a merged-cluster and other clusters in such a way, that the most similar clusters, μ and ν, are the ones that minimize the square-error

$$\Delta E^2_{\mu\nu} = \frac{n_\mu n_\nu}{n_\mu + n_\nu} \left(\vec{m}_\mu - \vec{m}_\nu\right)^2 \quad , \tag{2.3}$$

where m_μ is the 'centroid' of cluster ω_μ. This is done by defining the similarity between the clusters as

$$S(\omega_\mu \cup \omega_\nu, \omega_\gamma) = \frac{n_\mu + n_\gamma}{n_\mu + n_\nu + n_\gamma} S(\omega_\mu, \omega_\gamma) + \frac{n_\nu + n_\gamma}{n_\mu + n_\nu + n_\gamma} S(\omega_\nu, \omega_\gamma) - \frac{n_\gamma}{n_\mu + n_\nu + n_\gamma} S(\omega_\mu, \omega_\nu) \quad . \tag{2.4}$$

Our experience, as well as several comparative studies [1], have shown that Ward's method outperforms other hierarchical clustering methods. Therefore in this work we apply Ward's method wherever an example for agglomerative clustering methods is needed.

2.3 Super-Paramagnetic clustering

The *SPC* algorithm uses a granular ferromagnet as an analog device to cluster the data set. To each data point we assign a q-state Potts spin, namely a variable s which takes the values $1, \ldots q$. A *ferromagnetic* interaction $J_{ij} > 0$ is introduced between each pair $\langle i, j \rangle$ of neighboring points. The strength of the interaction decreases rapidly with increasing distance. Following Blatt *et al.* [12, 13] we choose to work with a Gaussian form of the interaction,

$$J_{ij} = \frac{1}{\hat{K}} \exp\left(-\frac{d_{ij}^2}{2a^2} \right) \quad , \tag{2.5}$$

where $\hat{K}$ is the average number of neighbors of each data-point, a is the average distance between neighboring points, and d_{ij} is the distance between the points x_i and x_j. The manner in which neighboring points are chosen is discussed in chapter 5.

A classification of the points is an assignment of values $\mathcal{S} = \{s_i\}$. For each assignment, a cost function $\mathcal{H}\left[\{s_i\}\right]$ is defined:

$$\mathcal{H}\left[\{s_i\}\right] = \sum_{\langle i,j \rangle} J_{ij} \left(1 - \delta_{s_i, s_j} \right) \quad . \tag{2.6}$$

This is just the Hamiltonian of a ferromagnetic Potts model.

At temperature $T = 0$, and low temperatures about it, such a disordered ferromagnet is in its ground state, where all spins are aligned. This is the *ferromagnetic phase.* At a very high temperature, the magnet is completely disordered, with each spin pointing in a random direction. This is the *paramagnetic phase.*

The manner in which the magnet goes from the ferromagnetic to the paramagnetic phases, as temperature is increased, depends on its structure. If the distribution of the spins is such that they form distinct clusters, then at some intermediate temperature the correlation between different clusters will be very weak, while within the clusters correlations are strong. At this temperature the system is in the *super-paramagnetic* phase.

The *SPC* algorithm constructs a hierarchical clustering by locating the temperature range of each of the super-paramagnetic phases, and identifying different clustering solutions at temperatures well within the different phases. These phases are identified using the *susceptibility* graph $\chi(T)$. The susceptibility, in general, measures the fluctuations in the size of the clusters at any given temperature. In order to formally define the susceptibility, one first has to define *total magnetization* $m(\mathcal{S})$. The total magnetization is a linear function of the number of data points with the most populated spin-value;

$$m(\mathcal{S}) = \frac{qN_{\max}(\mathcal{S}) - N}{(q-1)N} \tag{2.7}$$

where $N_{\max}(\mathcal{S}) = \max\{N_1(\mathcal{S}), N_2(\mathcal{S}), \ldots, N_q(\mathcal{S})\}$, $N_i(\mathcal{S})$ is the number of points with spin-value i, and N is the total number of spins. The susceptibility is related to the variance of the magnetization at a given temperature:

$$\chi(T) = \frac{n}{T}\left(\left\langle m^2 \right\rangle - \langle m \rangle^2\right) \ . \tag{2.8}$$

In principle, it is easy to identify the phase transition using the susceptibility since it has a peak whenever clusters break. This enables locating temperatures at which meaningful partitions are obtained.

At each relevant temperature, *i.e.* a temperature well within one super-paramagnetic phase, we calculate the correlation between all neighboring pairs $G_{ij} = \left\langle \delta_{s_i,s_j} \right\rangle$. This is done using a Swendsen-Wang Monte-Carlo simulation [16]. When G_{ij} exceeds a threshold θ, the two points x_i and x_j are linked. The cluster solution for this temperature is identified by locating the connected components induced by these links. Two points are assigned to the same cluster if they are connected by a "path" of neighboring pairs, whose correlations exceeds the threshold. The results are insensitive to the precise value of θ, as long as θ is well within 1 and $1/q$.

Since G_{ij} is a decreasing function of the temperature, each link that does not exist at some temperature T_1 cannot exist at a higher temperature $T_2 > T_1$. This ensures that the clusters created at different values of T form a hierarchy; every cluster found at $T = T_2$ is fully contained within some cluster created at $T = T_1$.

The *SPC* algorithm is summarized as follows [14]:

Step 1. Assign to each data point x_i a q-state Potts spin variable s_i.

Step 2. Calculate nearest-neighbor interactions J_{ij} according to eq. (2.5).

Step 3. Use the SW Monte-Carlo procedure, with the Hamiltonian of eq. (2.6) to calculate the susceptibility χ at a sequence of temperatures.

Step 4. Using the susceptibility, identify the super-paramagnetic regimes.

Step 5. Select a temperature in each super-paramagnetic regime, and calculate the correlation G_{ij} for each pair of points.

Step 6. Identify clusters as the connected components of a graph, whose vertices are the data-points, and whose edges $\langle i, j\rangle$ are such that $G_{ij} > \theta$.

The main feature of the *SPC* algorithm is its *collective clustering*. Whether two points belong to the same cluster or not is determine by a collective effect, not only by the distance between the points. The correlation function G_{ij} captures the density of points in the surroundings of i and j. This way, *SPC* is robust to noise, and reflects a natural notion of clusters.

Moreover, the *SPC* algorithm does not impose any assumption regarding the structure of the data. The hierarchical organization is observed by using a resolution control parameter, the temperature. If no clustering structure exists in the data, then none should be obtained by the algorithm: the magnet just goes directly from the ferromagnetic to the paramagnetic phase.

As described here, the *SPC* algorithm uses two external parameters: the value of the threshold θ, and the mean number of neighbors of each point (which we denote by K).

In most cases the clustering solution is robust to the choice of θ. However, there are cases in which the topology of the resulting dendrogram, *i.e.* the order in which clusters break, is affected by this choice. Furthermore, there are cases in which thresholding the correlation function yields un-natural results, which caused Wiseman *et al.* [14] to introduce a directed-growth heuristic. In another work [17] we review these problems, and suggest a manner in which they can be avoided.

The effect of the number of neighbors, on the other hand, is much more significant. Blatt *et al.* [12] suggested selecting neighboring pairs according to the *K-mutual neighbors criterion*. According to this criterion, the points i and j should be considered neighbors iff i is one of the K nearest points to j, and vice versa. This leaves a single parameter, K, to be determined.

This method was originally proposed in order to save computation time of the *SPC* algorithm by reducing the number of connections in the graph. It was then claimed that the performance of *SPC* is not sensitive to the value of K. As we discuss in chapter 5, there are cases in which a more careful choice of K should be made. In this work we use validation methods to identify a good choice of this parameter, while in another work [17] we show how making this choice can be avoided completely.

Chapter 3

Cluster Validity

Cluster analysis is the process of searching for *natural* groups in a data set. As stated, this problem is far from being well-defined, since the so called "naturality" of a particular partition is pretty much in the eyes of the beholder. For example, one may expect clusters to be compact and dense structures in space; in many cases, however, the true components of some spatial distribution may be neither compact nor dense.

Most clustering algorithms invoke some heuristic procedure to identify clusters of data points. In practice, in nearly all cases there are some underlying (many times hidden) assumptions regarding what natural clusters should look like. If the data under investigation indeed satisfy these assumptions, the corresponding algorithm may do a good job identifying these "natural" clusters. For example, many parametric (such as EM [18]) as well as non-parametric (such as deterministic annealing [19]) methods assume that clusters should be compact. Others (such as Valley-seeking [7], and some graph-theory based algorithms [1]) identify clusters by looking for separating valleys in the data density distribution.

> The concept of cluster validation refers to methods and indicators which attempt to evaluate the results of a cluster analysis.

This evaluation should be quantitative and objective, if it is to have any meaning. The aim of cluster validation is therefore to evaluate objectively the quality of an answer to a subjective question.

Cluster validation has two major uses. First, one wishes to evaluate the statistical significance of the results obtained by the clustering analysis. Such a figure of merit may be used in order to determine how significant is the claim, that two points reside in the same cluster. Many times one finds that applying several clustering algorithms to the same data yields different results. This should be no surprise: as already mentioned, different clustering methods interpret differently the notion of "natural" clustering, and therefore look for different partitions, and may also have different sensitivity to noise (inherently present in the data). One then wishes to use that method, whose result is most reliable statistically.

A second use of cluster validation is to optimize the choice of external parameters on which the algorithm may depend, such as the expected number of clusters (*e.g.* EM) or the number of "neighbors" of any point (*e.g.* Valley-Seeking, *SPC*). Cluster validation is then used to determine the optimal parameter to be used, *i.e.* the value which yields the most valid clustering solution.

An important example is the case of the parameter that controls resolution. Many clustering methods produce a dendrogram, which can be viewed as several different solutions, at different levels of resolution, ranging from a single cluster that contains all points to N clusters with a single data-point in each. In many cases, however, one is interested only in a single "best" or "most natural" clustering, and there is no figure of merit that helps to determine the corresponding resolution, at which one should "cut" the dendrogram. Even if such a figure of merit does exist, its reliability may be questionable. In this case, the question of the validity of each resolution level should be answered externally.

Given a validity measure, one may use it to compare several possible solutions; or just use it to evaluate a given solution. In the later case, some statistical distribution of this figure of merit has to be determined, in order to evaluate the statistical significance of the clustering solution in question. This distribution is usually obtained using Monte-Carlo methods [1].

3.1 Cluster Validity - a simple example.

Before presenting our approach to cluster validation, let us briefly present a simple example, that demonstrates the problems we are trying to solve.

Consider the probability distribution given in figure 3.1: the probability is uniform inside the dark area, and vanishes outside. The dumbbell-shaped distribution can be described as two circular regions connected by a bridge. Clearly, the number of clusters suggested by this probability function is a subjective matter. Different samples of 600 points are drawn from this probability distribution.

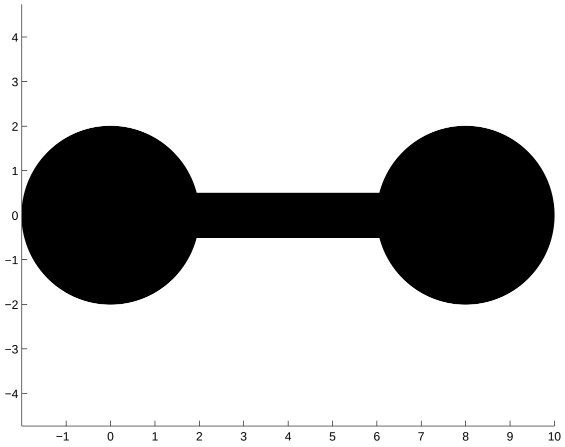


Figure 3.1: The probability probability density of our example is uniform in the dark area, and vanishes in the white area.

A typical sample is shown in figure 3.2. We would like to perform clustering analysis, in order to estimate the number of components of the distribution function. Remember that clustering analysis is an unsupervised approach, so we must assume that nothing is known about this function.

We approach this problem with three different clustering algorithms. First, we apply the *Deterministic Annealing (DA)* algorithm [19]. This algorithm is given a maximal number of expected clusters, C. If only a fewer number of cluster is identified, some of the resulting clusters are expected to be degenerate.

The cost DA assigns to any partition is based on the distance of each point from the centroid of the cluster to which it is assigned. In other words,

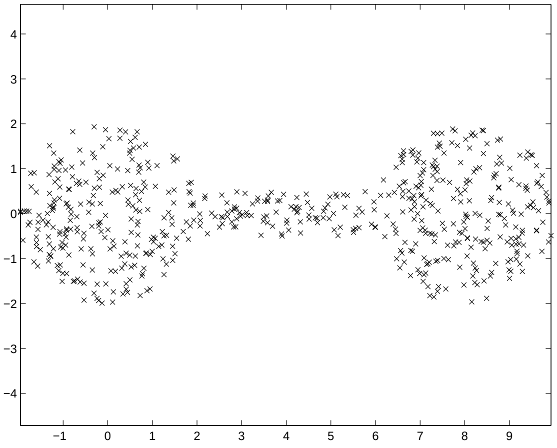


Figure 3.2: A typical sample. 600 points were drawn from the probability density of figure 3.1.

DA assumes that the clusters can be reasonably described in terms of their centroids. This approach is suitable if one seeks a solution that will be used for data compression.

If one views the probability function of figure 3.1 as a single cluster, then clearly this cluster is far from comprising a tight distribution about a centroid. Indeed, as we start DA of with the maximal number of clusters set to $C = 2$, the algorithm breaks the data into two sub-clusters, as shown in figure 3.3(a). Running the algorithm with $C = 3$ results in three clusters, shown in figure 3.3(b).

The second algorithm we apply to this data set is Ward's algorithm. This being a hierarchical clustering method, it's results are best presented by a dendrogram (figure 3.4). Like all agglomerative algorithms, Ward's method has to break the data into clusters. Indeed, the first partition breaks the data roughly in the middle of the bridge. However, the level of the dendrogram, if any, at which the resulting partition is reliable, is not known.

Finally, we let *SPC* have its way with the data. The sizes of the largest clusters, as function of temperature, as well as the susceptibility curve, are given in figure 3.5. Only a single transition is observed, at $T = 0.095$. The analogous Potts magnet moves directly from the ordered (ferromagnetic) phase (a single cluster) to the completely disordered (paramagnetic) phase

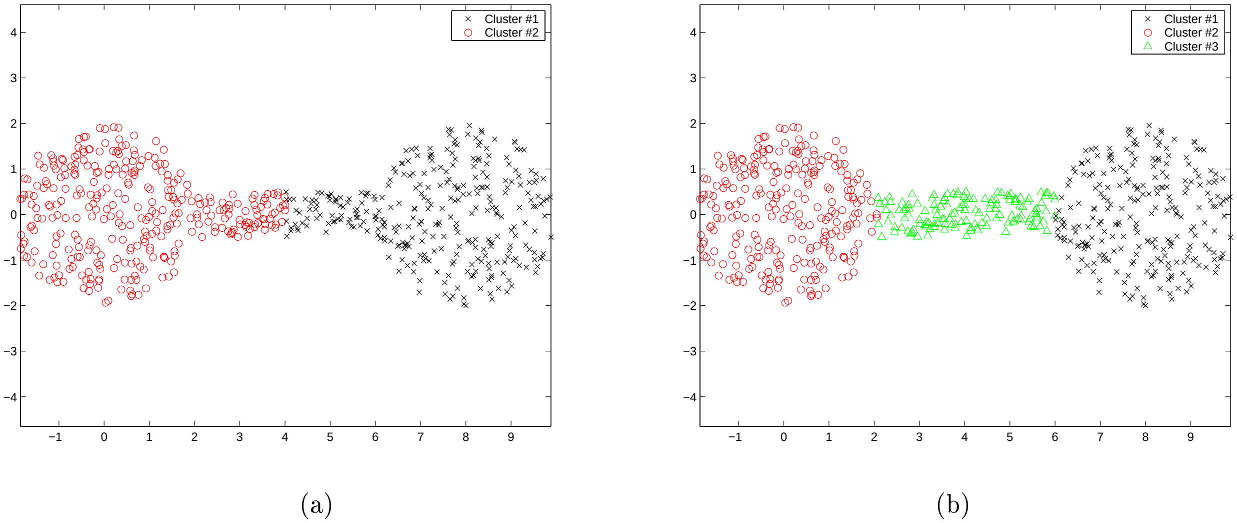


Figure 3.3: Clustering solution according to the Deterministic Annealing Algorithm, with *(a)* $C = 2$ and *(b)* $C = 3$ clusters.

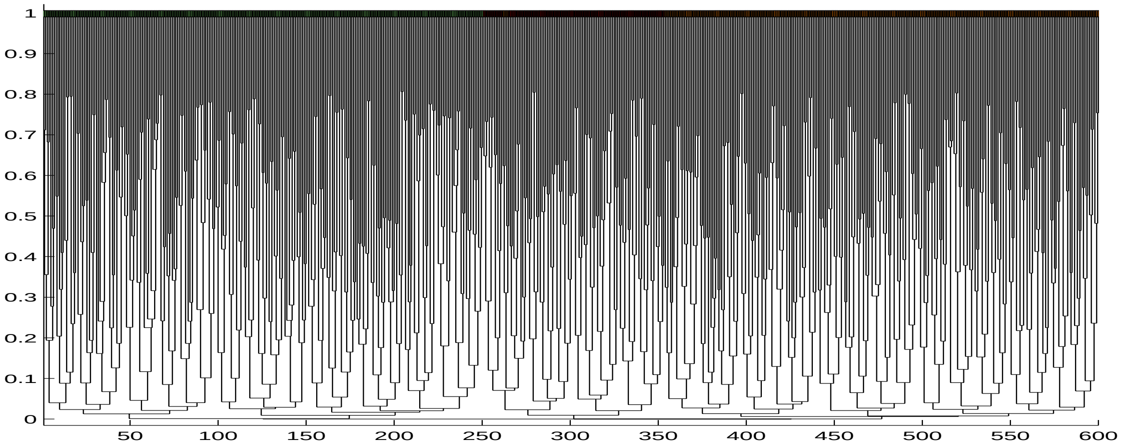
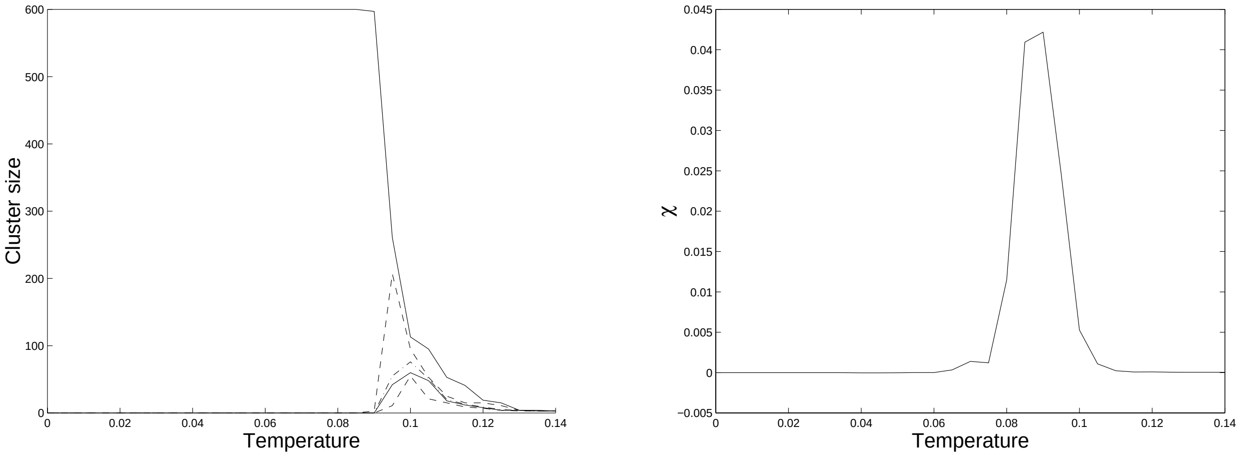


Figure 3.4: Clustering solutions according to Ward's algorithm.

(many small "clusters"). *SPC* identifies correctly the absence of cluster structure in the data.



(a) (b)

Figure 3.5: Cluster size *(a)* and susceptibility curve *(b)*, according to *SPC* . No super-paramagnetic phase is observed.

A somewhat less typical sample, drawn from the same probability function, is given in figure 3.6. One can observe that due to fluctuations, a wide gap has appeared at the middle of the bridge. Indeed, in this case *SPC* identifies a super-paramagnetic phase with two clusters (figure 3.7).

Our aim is to identify the "correct" number of components in the underlying probability function. We saw that DA, Ward and *SPC* provided us with different results for the same data (figure 3.2). The reason for this is that the assumptions made implicitly by DA are not suitable for the data, while Ward (like any agglomerative algorithm) always breaks the data into clusters (even when none realy exists). The unsupervised learner, however, remains puzzled, and would like to have a way to choose which of the different answers is most suitable for the data.

Furthermore, DA gave different answers for different values of the external parameter C. We also observed that a somewhat untypical sample, taken

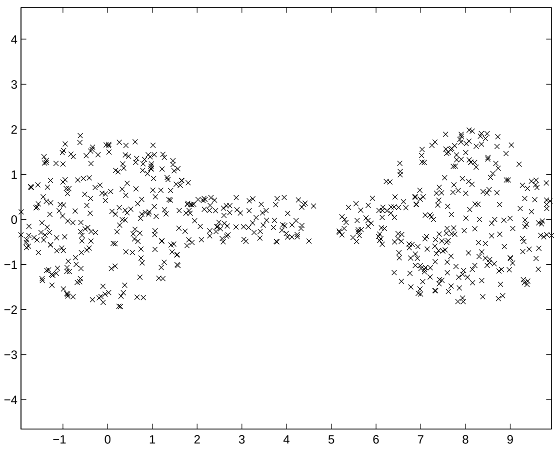
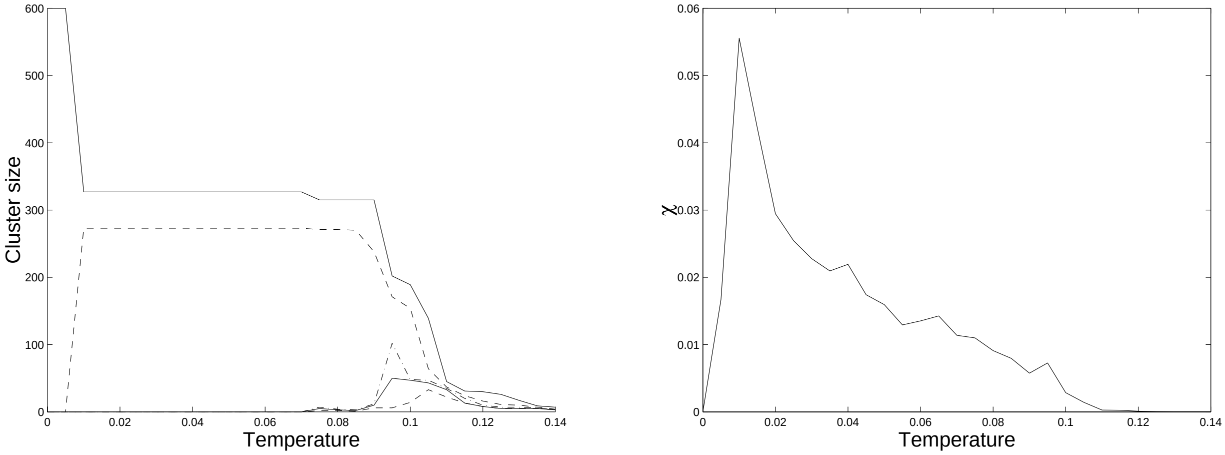


Figure 3.6: A less typical sample. 600 points were drawn from the probability density of figure 3.1. A gap in the data can be observed in the middle of the bridge.



(a)

(b)

Figure 3.7: Cluster size *(a)* and susceptibility curve *(b)* for the second sample, according to SPC . A super-paramagnetic phase is observed. In this phase, two cluster are identified, consisting of points on different sides of the gap.

from the same distribution, can lead to erroneous results. In most cases, of course, it is not trivial to observe that the sample is, in some sense, untypical. Moreover, it may be that this sample, typical or not, is all one has to work with. The user faces several possible clustering solutions and would like to know which of them is less likely to be an artifact caused by some unlikely fluctuation. Cluster validation should supply the answers to this kind of questions.

Chapter 4

Resampling approach to Cluster Validity

Imagine that a data set is drawn from a probability distribution, which is a mixture of several density functions. One may state that the aim of clustering is to identify the different components of the distribution, in the sense that every cluster should correspond to one of the components.

The ability to succeed in this task depends on several conditions. First, there must be some separation in the data space between the high density regions of the different components. In case of substantial overlap, no distinction between the density functions is possible without making strong assumptions about the structure of the data. Second, there should be enough representatives of each component in order to recover the underlying structure.

A third condition, which is rather difficult to test, is that the data set represents correctly the underlying distribution. For example, imagine that in the example of the previous section, an improbable sample is drawn, with not even a single data point in the bridge. Clearly in this case no clustering analysis will recover the single-cluster structure correctly.

Let us say that a scientist is investigating mice, and comes to suspect that there are several types of them. For whatever reason, this scientist believes she can distinguish between the types by measuring only the weight of the mice. She therefore weighs all the mice in her lab, looks for clusters in this one-dimensional data, and finds two clusters. This result can be interpreted in two ways. The first is that there are indeed two types of mice. We can say, that in this case the two-cluster solution is “natural” to this problem. However, it may also be that there is only one type. The reason for the apparent two-cluster solution in this case is that there were not that many mice in the lab, and it so happened that for some range of weights there was no representative mouse. We call this a sampling artifact. Cluster validity is interpreted in this section as the need to distinguish between natural clusters and sampling artifacts.

In order to cope with the noise which arises from sampling the data set, one may wish to work with several different samples, and compare the results of clustering analysis of each one of them. In practice, one cannot afford, or may not have available, a large sample set. To overcome this problem, we may use resampling methods to obtain different samples from the original set.

We expect that breaks in the data, which are due to the genuine structure of the underlying probability distribution, will be present in most of the resamples. On the other hand, breaks or gaps that are due to the sampling noise should appear only in a small fraction of the resamples. This would enable us to identify correct clustering solutions, and leave out those partitions which are due to sample artifacts.

4.1 Quantitative analysis: A one-dimensional example

To give a quantitative expression to the qualitative statements made above, we analyze a simple one-dimensional case, and show that indeed natural clustering separation is relatively stable under resampling.

Let us look at a one dimensional data set which is constituted of points selected from a uniform distribution, characterized by the mean distance between neighboring points, $1/\lambda$. If our clustering results reflect the underlying distribution from which the data were selected, then clearly a ’correct’ answer should identify only one cluster in this data set.

Consider a simple nearest-neighbor clustering algorithm, which assigns two neighboring points to the same cluster if and only if the distance between the two is smaller than a threshold a, where a is a parameter of the algorithm. Clearly, a controls the resolution of the clustering: for very small a, no two points belong to the same cluster, and the number of clusters is just the number of points; for very large a, larger than any nearest neighbors distance, all pairs of neighbors are assigned to the same cluster, and hence all points reside in one cluster.

We would like to validate the clustering solution for the full data set, with a given value of a, using resampling. The resampling scheme we use is to decide independently for each data point whether it is kept in the resample (with probability f), or is discarded (with probability $1 - f$). We call f the *dilution factor*. Length scales of the original problem get rescaled by the resampling procedure by a factor of $1/f$, so that the mean distance between neighboring points in the resamples is roughly $1/\lambda f$. Clustering is therefore performed on the resample with a rescaled threshold parameter $a' = a/f$.

The nearest neighbor distances between the points of our sample are distributed according to the Poisson distribution,

$$P(s) = \lambda e^{-\lambda s} ds. \tag{4.1}$$

Let us choose the parameter a, and assume that the nearest-neighbor algorithm falsely identifies a break into sub-clusters, *i.e.* there exists some nearest-neighbor distance b which is larger than a. We would like to estimate the probability that this break into sub-clusters will also appear in a resample.

We therefore identify, after the resample, two neighboring points closest to the gap, which reside on different sub-clusters in the original solution. We ask what is the probability that the distance between the two after resampling exceeds $a' = a/f$, in which case the division into two sub-clusters appears at the same place. In order to answer this questions, we should find out what is the probability that the distance between the two is constructed of m distances of the original sample, and then find out what is the probability for such a combined distance to be larger than the new threshold.

In order for the distance under consideration to be constructed of m original distances (on top of the one which created the gap), m points of the sample must not survive the resampling, while the $(m + 1)$ point should survive. The probability for such an event is $f(1 - f)^m$.

The combined distance in this case is distributed like the sum of m Possionian random variables, that is with probability distribution

$$P_m(S) = \frac{(\lambda S)^{m-1}}{(m-1)!} e^{-\lambda S}. \tag{4.2}$$

From this we conclude that the probability for the total distance between the two alleged sub-clusters to be greater than a/f, is just the probability for the sum of the m-distances to be greater than $a/f - b$, that is

$$B_m \equiv \frac{1}{(m-1)!}\Gamma(m, \alpha/f - \beta)), \tag{4.3}$$

where we have introduced the dimensionless variables $\alpha = \lambda a$ and $\beta = \lambda b$. $\Gamma(n, z)$ is Euler's incomplete Gamma function,

$$\Gamma(n, z) = \int_z^\infty e^{-t} t^{n-1} dt, \tag{4.4}$$

except that in our convention, $\Gamma(n, z < 0) = \Gamma(n, 0) = (n-1)!$, so $B_m = 1$ for $\alpha \leq f\beta$.

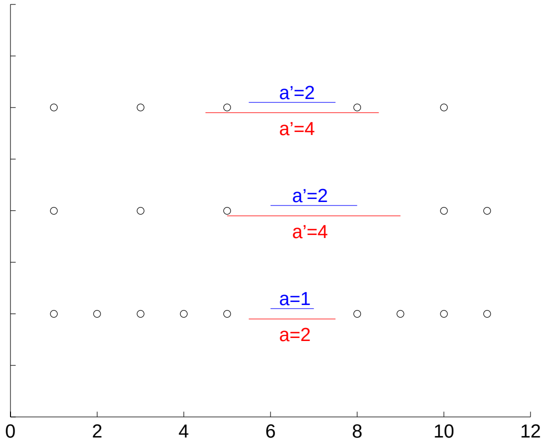


Figure 4.1: This figure shows a one dimensional data set (bottom) and two resamples with dilution factor $f = 1/2$ (middle and top). We consider two values of the resolution parameter: $a = 1$ and $a = 2$. The gap in the original sample is $b = 3$, so two clusters are identified with both resolutions. In the resamples we use a rescaled resolution parameter, $a' = 2a$. For $a = 1$, *i.e.* $a < b/2$, the gap survives any resample, so $P_{gap}(a < bf) = 1$. For $a = 2$, however, the gap may survive (middle) or may not (top), and hence $P_{gap}(a > bf) < 1$.

Altogether, we find that the probability for a gap to appear in the same place in the resample is given by

$$P_{gap} = \sum_{m=1}^{\infty} \frac{f^2(1-f)^{m-1}}{(m-1)!}\Gamma(m, \alpha/f - \beta), \tag{4.5}$$

where we took the upper limit of the summation to be infinite assuming that sample size is large, and noticing that the summation terms decay very fast. This probability function and the arguments given above are explained in figure 4.1 and its caption.

The probability P_{gap} is plotted in figure 4.2 against the dimensionless variable $\alpha = \lambda a$, for of $\beta = 5$. In this figure we have taken the dilution factor f to be either 2/3 or 1/2. For small values of α (where $\alpha < f\beta$, the probability for the gap to reappear in the resample is unity. As a gets larger, this probability declines.

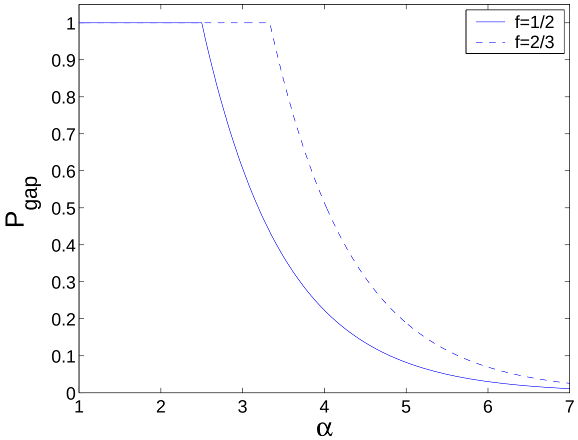


Figure 4.2: The probability that a gap of size $b = 5/\lambda$ in the bulk of a cluster will survive resampling, as a function of the dimensionless parameter $\alpha = \lambda a$. Dilution factors $f = 2/3$ and $f = 1/2$ were used.

Figure 4.3 shows P_{gap} again, but here we take $\beta = \alpha$. The fast decay of this probability demonstrates the fact, that gaps which are much larger than the mean distance are very unlikely to survive resampling at the relevant resolution.

Similarly, we may look at two points, whose distance from each other is smaller than a, so $\beta < \alpha$. These two points reside in the same cluster at this

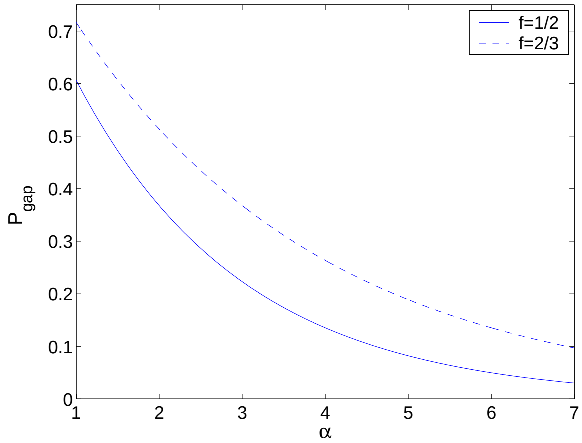


Figure 4.3: The probability of a gap of size $b = a$ in the bulk of a cluster to survive resampling, as a function of the dimensionless parameter λa. Dilution factors $f = 2/3$ and $f = 1/2$ were used.

level of resolution. The probability that this is the case after resampling is the probability that the new distance is constructed from m original distances (given by (4.2)), times the probability that the sum of distances is smaller than $a/f - b$:

$$P_{int} = \sum_{m=1}^{\infty} \frac{f^2(1-f)^{m-1}}{(m-1)!}\gamma(m, \alpha/f - \beta). \tag{4.6}$$

Here $\gamma(n, z)$ is the other incomplete Gamma function,

$$\gamma(n, z) = \int_0^z e^{-t}t^{n-1}dt, \tag{4.7}$$

and we take $\gamma(n, z < 0) = \gamma(n, 0)$. This function is getting close to unity very fast for $\alpha > \beta$, as we learn from figure 4.4, where P_{int} is drawn for the case of $\beta = 5$. This behavior marks the stability of the correct solution.

4.2 Using resampling to determine relevant resolutions

Let us now allow another uniformly distributed cluster in this problem, separated from the other cluster by some distance Δ. Assume that $\Delta\lambda \gg 1$,

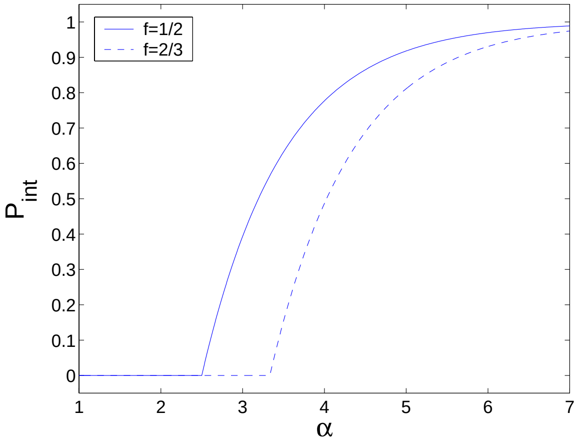


Figure 4.4: The probability P_{int} for a gap of size $\beta = 5$, which for $\alpha > 5$, does not break the cluster in the original resample, not to do so in the resample. Here we used the dilution factors $f = 2/3$ and $f = 1/2$.

such that the two clusters are easily separated. Clearly, if $\Delta > a/f$, this gap will survive the resampling. Making a large enough to be comparable with Δ, so that the gap between the clusters may be unstable, implies that our dimensionless variable λa is very large. From P_{gap} of eq. 4.5 we learn that in such a case no other crack in the data (inside either cluster) can be even remotely stable.

More generally, we may identify five different regimes of the threshold parameter a :

1. For very small a, practically any point forms its own cluster. This solution is stable to resampling.

2. For a comparable with $1/\lambda$, we find some cracks inside each cluster. As we've seen, this solution is very unstable to resampling.

3. As a gets much larger, but still substantially smaller than Δ, the clustering solution identified by the nearest-neighbor algorithm is the true solution. This solution is once again stable to resampling.

4. The stability of the solution declines again as a becomes comparable with Δ.

5. Finally, as a becomes very large, all data points reside in the same cluster. This solution is again very stable.

There are therefore three regimes of the resolution parameter, where clustering solutions are robust against resampling. First, there are the low and high ends, which should be considered trivial. Than, there may be other stable regimes, whose existence marks other relevant scales in the problems.

4.3 Figure of merit for stability under resampling

We turn now to define a score $\mathcal{M}$ that reflects the stability of a clustering solution under resampling.

First, let us define the clustering assignment matrix $\mathcal{T}$,

$$\mathcal{T}_{ij} = \begin{cases} 1 & \text{points } i \text{ and } j \text{ belong to the same cluster} \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

Our figure of merit is based on comparing two such matrices, $\mathcal{T}$ and $\mathcal{T}^{(n)}$, where the first corresponds to the clustering of the original sample, and the second to clustering the nth resample.

We are now ready to define our figure of merit,

$$\mathcal{M} = \ll \delta_{\mathcal{T}_{ij},\mathcal{T}_{ij}^{(m)}} \gg \tag{4.9}$$

where the averaging implied by the $\ll \cdot \gg$ notation is over all neighboring pairs which survived the resample, and then over the resamples. Clearly, $0 \leq \mathcal{M} \leq 1$, with $\mathcal{M} = 1$ for perfect score.

If our clustering algorithm depends on some external parameters, then so does the assignment matrix. It is than clear that $\mathcal{M}$ depends on the same set of parameters. One must take care to define the parameters in such a way, that it would be reasonable to apply the algorithm with the same set of parameters both when clustering the original sample and any of its resamples. For example, in the nearest-neighbor algorithm described earlier, specifying the threshold in terms of the mean nearest-neighbor distance would enable using the same value for the original sample as well as for the resample.

We can state the role of $\mathcal{M}$ formally: *given a clustering algorithm $\mathcal{A}(\alpha)$, which depends on a set of external parameters α, the reliability of the clustering result can be assessed by $\mathcal{M}(\alpha)$.*

4.4 Quantitative insight - cont.

We go back now to the one dimensional model and the nearest-neighbor clustering algorithm, and examine the behavior of the function $\mathcal{M}$ in this case.

Let us choose two pairs of neighboring points. The first pair resides inside one of the two components of the distribution. Let us denote its points i and j, and the distance between them b. We can write the contribution of this pair to $\mathcal{M}$ in the following way:

$$\begin{aligned} &P(\mathcal{T}_{ij}=1)P(\mathcal{T}'_{ij}=1|\mathcal{T}_{ij}=1)+P(\mathcal{T}_{ij}=0)P(\mathcal{T}'_{ij}=0|\mathcal{T}_{ij}=0) \\ &=P(b<a)P_{int}+P(b>a)P_{gap}, \end{aligned} \tag{4.10}$$

where $\mathcal{T}$ is the assignment matrix for the clustering solution of the original sample, and $\mathcal{T}'$ the assignment matrix of a resample. In order to determine the typical behavior of this quantity, we average it over all possible values of b. For values of b which are smaller than a, no break appears in the original sample, and therefore we consider the probability that none would appear in the resample. For values of b which are larger than a, a break into subclusters does appear in the sample, and therefore in that case we consider the probability that the same would happen in the resample. Finally, we combine these two regimes of b, and have

$$A(\alpha)=\int_0^\alpha e^{-\beta}P_{int}\,d\beta+\int_\alpha^\infty e^{-\beta}P_{gap}\,d\beta \tag{4.11}$$

where we have used again the dimensionless variables $\alpha=\lambda a$ and $\beta=\lambda b$.

The two points which make up the second pair, k and l, are on two different sides of the gap Δ between the two distribution components. Here we have $\mathcal{T}_{kl}=\Theta(a-\Delta)$, where $\Theta(x)$ is the unit step function. The contribution of this pair to $\mathcal{M}$ is therefore given by

$$B_\delta(\alpha)=\left\{\begin{array}{ll} P_{int} & \alpha\geq\delta \\ P_{gap} & \alpha<\delta \end{array}\right. , \tag{4.12}$$

where we introduced another dimensionless variable, $\delta = \lambda\Delta$. We note that this contribution is non-analytic, due to the non-analyticity of the distribution function, caused by the abrupt switch from P_{int} to P_{gap}.

In order to estimate $\mathcal{M}$, we should average over the contribution of all pairs. Of course, there are many more pairs inside the two components then ones which cross the gap between them. Nevertheless, we weigh here the two types equally, since this is more typical to higher dimensional cases. In the numerical experiments that follows, we do this by counting all pairs (not just neighboring ones). Our final expression for $\mathcal{M}$ is therefore

$$\mathcal{M}(\alpha) = \frac{1}{2}\left[A(\alpha) + B_\delta(\alpha)\right]. \tag{4.13}$$

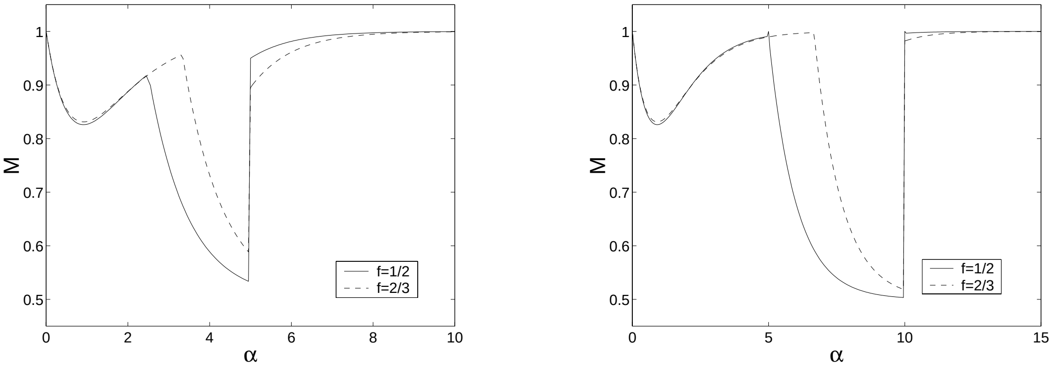
Of course, $\mathcal{M}(\alpha)$ depends implicitly on the dilution ratio, f. This function is plotted, for both $f = 1/2$ and $f = 2/3$, in figure 4.5, where we have assumed the inter-cluster distance $\Delta = 5/\lambda$. A clear peak can be observed in both curves at $\alpha \simeq 2.5$ and $\alpha \simeq 3.3$, respectively. These peaks correspond to the most stable solution. Looking at the profile of $\mathcal{M}$ as a function of α, one can clearly observe the five regimes discussed in section 4.2.

4.5 Numerical Experiments in One Dimension

We show now that the score $\mathcal{M}$ indeed behaves in the way we anticipated in section 4.2, using several numerical experiments, with different sets of one dimensional data.

First, we used the nearest-neighbor algorithm on data similar to that described in our calculation, namely two uniformly distributed clusters, with separation ten times as large as the mean nearest neighbor distance, $\Delta\lambda = 10$. Our full sample is of size $N = 200$, and each resample is of size 130 (*i.e.* $f \approx 2/3$). We created 100 resamples, and applied the geometric clustering procedure for each one of them.

We express again the threshold parameter by the dimensionless variable α, such that the threshold is given by α/λ, where again λ^{-1} is the mean nearest-neighbor distance. Using this notation, we apply the algorithm with the same value of the parameter α both for the full sample and for its resamples.



(a) (b)

Figure 4.5: Mean behavior of $\mathcal{M}$ as a function of the geometric threshold, according to equation (4.13), for two clusters. The function is evaluated for the inter-cluster distances (a) $\Delta = 5\lambda$ and (b) $\Delta = 10\lambda$, with dilution parameters $f = 1/2$ and $f = 2/3$.

Our figure of merit is calculated for different values of α, as shown in figure 4.6. The peak between $\alpha \approx 4$ and $\alpha \approx 7$, corresponding to the correct clustering solution, is clearly identified.

We next turn to the case of three clusters. Here we can distinguish between two cases: the first, when the distance between the left cluster and the middle one is considerably smaller than the distance between the right cluster and the middle one; $\Delta_1 = 10\lambda^{-1}$, $\Delta_2 = 20\lambda^{-1}$. In this case, a hierarchical clustering solution which first breaks the whole data into two clusters and then, as α increases, breaks one of the clusters into two sub-clusters, can be considered as correct. Our figure of merit for this case is shown in figure 4.7(a), and indeed - the two solutions (of two and three clusters), observed at $\alpha = 7$ and $\alpha \simeq 12$ respectively, can be identified as the correct ones.

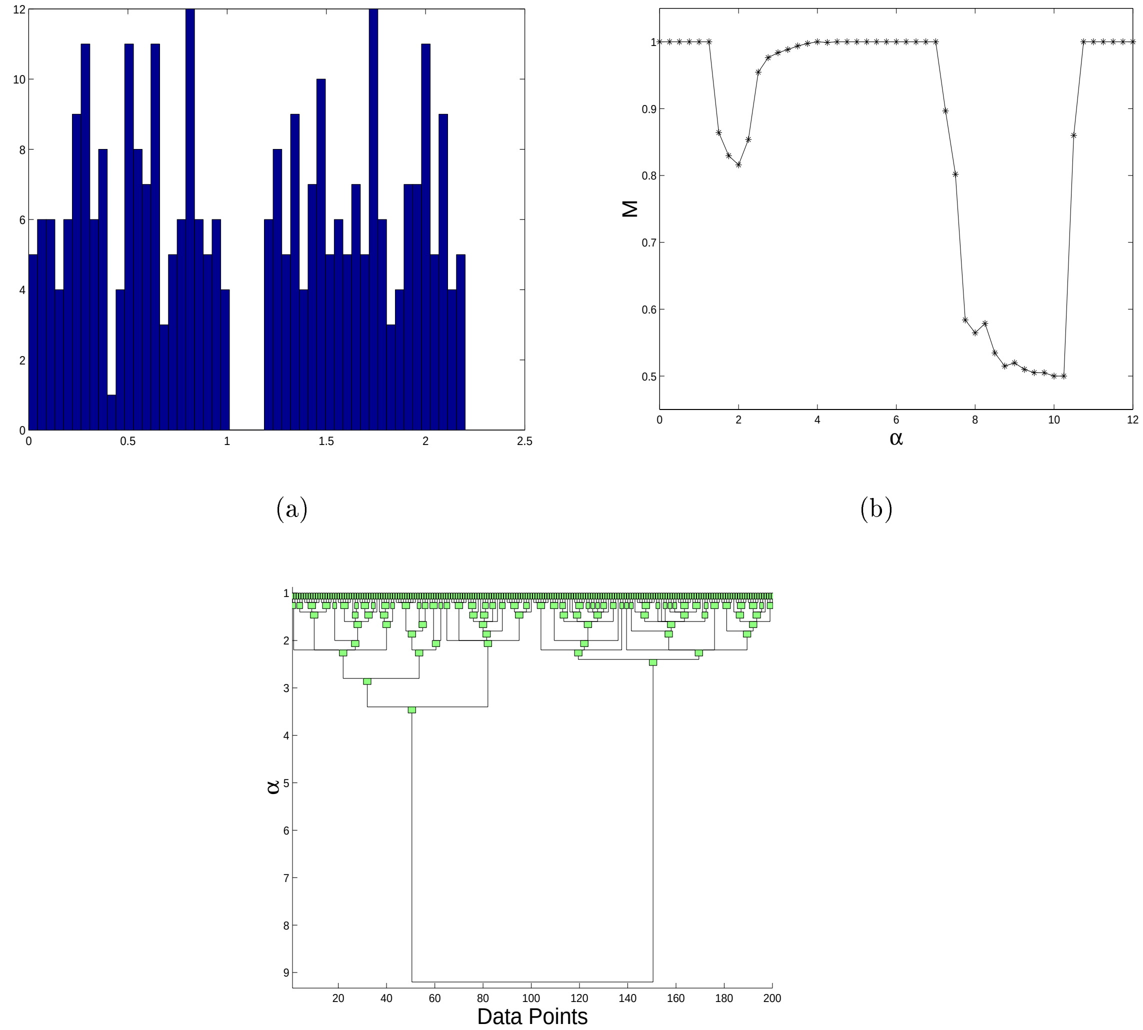
The second case is the one where the separations between adjacent clusters are more or less the same. In this case, there is only one correct solution, which identifies three clusters. The curve obtained for $\mathcal{M}(\alpha)$ (fig. 4.7(b)) agrees with this statement.

When trying to apply the same approach to noisier data, we found that the noise may be comparable with the size of the peak, and may wipe it out completely. In order to deal with this effect, we limit the calculation of $\mathcal{M}$ to points which belong to clusters larger then $\sqrt{N}$, where N is the size of the sample, *i.e.* 'macroscopic' clusters.

The same approach was taken for data built of 3 Gaussian clusters, equally separated, with $\Delta = 20\lambda^{-1}$. The results are show in figure 4.8. Values of α near the prominent peak correspond to the correct solution (of three clusters).

4.6 The temperature is a resolution parameter

In the next sections we apply the resampling scheme described above to data of higher dimension, using the *SPC* algorithm. We would like to see the connection between the resolution parameter α of the geometric model, and the parameters of *SPC* . To this end we consider a one dimensional random



(a)

(b)

(c)

Figure 4.6: Resampling results for a one-dimensional data set. 200 points were chosen uniformly from two clusters, separated by $\Delta\lambda = 10$. Histogram of the data is given in (a). We performed 100 resamples of 130 points, (*i.e.* $f \approx 2/3$) to calculate $\mathcal{M}$. In (b) we plot $\mathcal{M}$ as a function of the resolution parameter α. The peak between $\alpha \approx 4$ and $\alpha \approx 7$ corresponds to the correct two cluster solution, as can be seen from the dendrogram shown in (c).

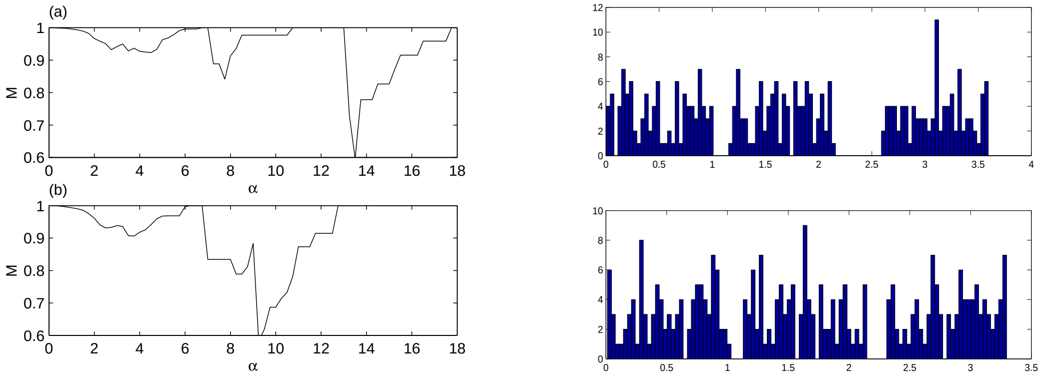


Figure 4.7: $\mathcal{M}$ as a function of the geometric threshold, for two data sets, constructed from three clusters, with non-equal (a) and equal (b) separation. Histograms of the two data sets are given on the right. The two cluster solution, seen at an intermediate resolution, should be considered correct only in the first case.

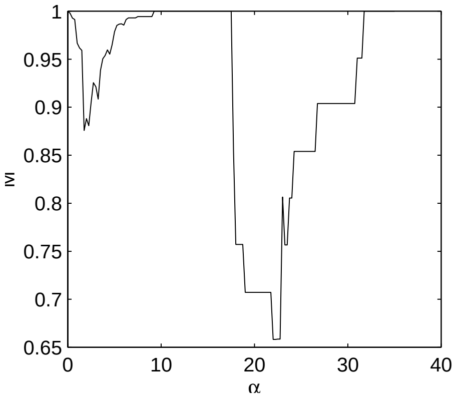


Figure 4.8: $\mathcal{M}(\alpha)$ for three equally spaced clusters. The shortest distance between points of different clusters is $\Delta \approx 20\lambda^{-1}$. The points of each cluster are chosen with normal distribution about different means. 20 resamples were performed with dilution factor $f = 2/3$. Only points in "macroscopic" clusters were considered.

Ising model, with coupling constants defined, as in *SPC* , by

$$J_{ij} = \frac{\sqrt{\lambda}}{2}\exp(-\frac{\lambda^2 d_{ij}^2}{2}), \tag{4.14}$$

where d_{ij} is the distance between neighbor points on a line, sampled from some distribution.

The clustering assignment is such that two nearest neighbor points are assigned to the same cluster if the correlation G_{ij} is larger then some threshold θ.

We then notice that

$$\begin{aligned} \mathcal{T}_{i,i+n} &= \prod_{j=0}^{n-1} \Theta(G_{i+j,i+j+1} - \theta) \\ &= \prod_{j=0}^{n-1} \Theta(\tanh \frac{\beta}{2} e^{-\frac{\lambda^2 d^2_{i+j,i+j+1}}{2}} - \theta) \\ &= \prod_{j=0}^{n-1} \Theta(\alpha - \lambda d_{i+j,i+j+1}) \end{aligned} \tag{4.15}$$

where $\alpha \equiv \sqrt{-2\log\frac{2}{\beta}\tanh^{-1}\theta}$.

This is a one-to-one correspondence between the geometrical resolution parameter α and the inverse temperature β. A higher temperature (smaller β) means a higher resolution (larger α).

We also expect the resolution of the physical model to depend on the range of the interactions, so we expect to find similarity between the dependence of $\mathcal{M}$ on the number K of neighbors, to the dependence on α. This dependence is not manifested in these one dimensional models, where the concept of nearest-neighbors is more strict.

4.7 Resampling and the *SPC* algorithm

In previous sections we have shown how resampling can be used to determine relevant resolution at which a problem should be clustered, and have made the connection between the temperature variable of the *SPC* algorithm and the resolution parameter.

SPC provides a hierarchical clustering solution. In section 2.1 we mentioned that one of the problems of hierarchical algorithms is that they often do not provide a method to identify which levels of the hierarchy are relevant to the problem. The *SPC* algorithm does enable locating the meaningful levels of the tree by the susceptibility curve. The question, which of these levels indeed reflects the underlying structure of the data, is still open. This question is approached here by the method of resampling.

Again, we demonstrate this behavior with a toy problem which can be visualized. This is a two-dimensional data set, shown in figure 4.9. The angular coordinate is uniformly distributed, $\theta \sim \mathrm{U}[0, 2\pi]$. The radial coordinate is normal distributed, $r \sim \mathrm{N}[R, \sigma]$ around three different radii R. The outer "ring" ($R = 4.0$, $\sigma = 0.2$) consists of 800 points, and the inner "rings" ($R = 2.0, 1.0$, $\sigma = 0.1$) consists of 400 and 200 points, respectively. The partition which best represents this distribution function is, of course, the one which identifies the three "rings" as three different clusters.

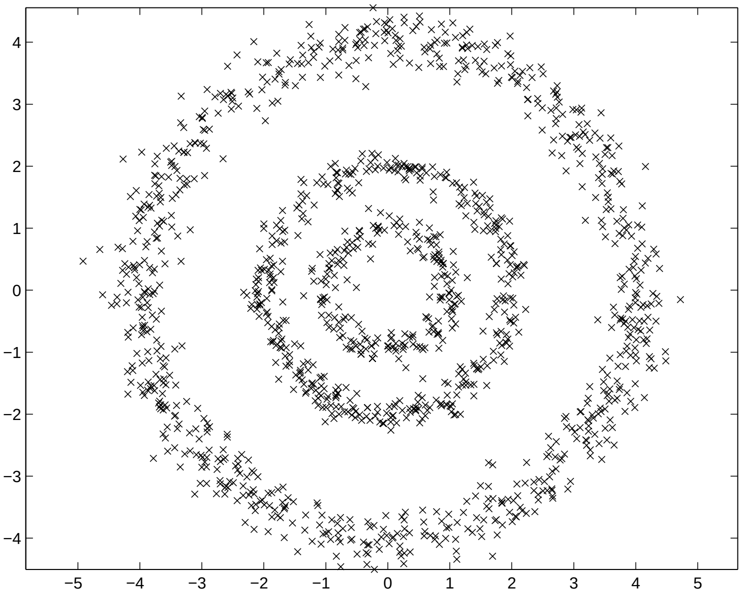
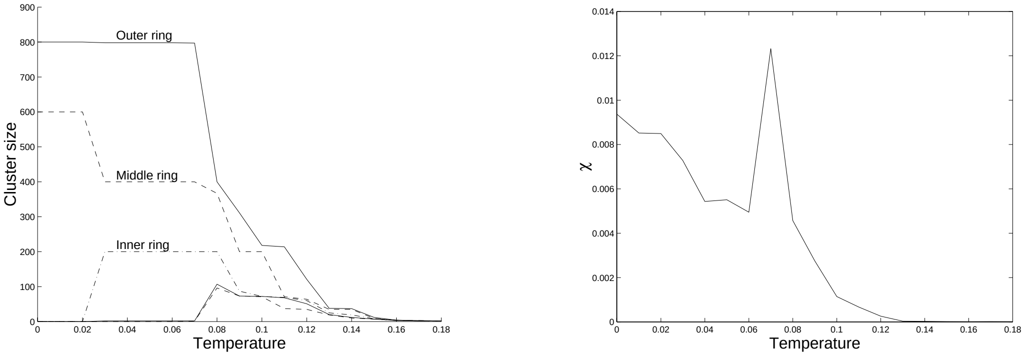


Figure 4.9: The three-ring problem. 1400 points are sampled from three distributions described in the text.

The results of applying the *SPC* algorithm to this data set is given in figure 4.10. We identify here three levels of resolution, apart from the two



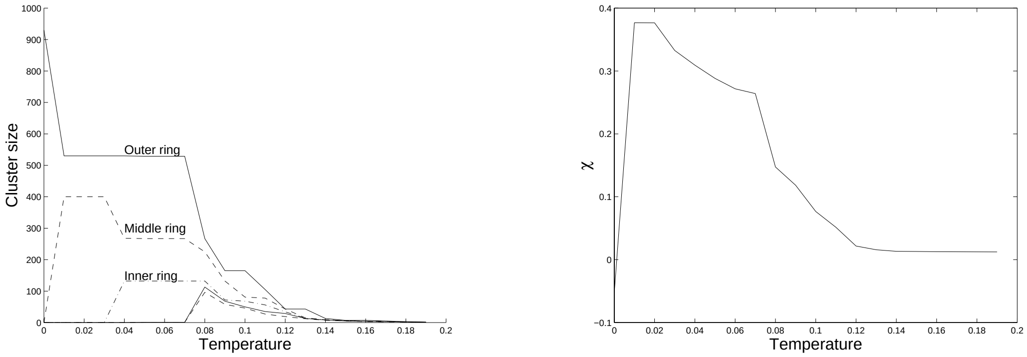
(a) (b)

Figure 4.10: *SPC* results for the full data set. figure *(a)* shows sizes of the largest clusters as a function of the temperature. Figure *(b)* shows the susceptibility curve.

trivial ones: *(i)* Two large clusters, one of them corresponding to the outer ring ($T \simeq 0.01$); *(ii)* Three clusters, corresponding to the three rings ($T = 0.03 - 0.08$); and *(iii)* small clusters, of tens of points in each.

We performed 20 different resamples from this toy data set, with a dilution factor of $f = 2/3$, and find it easy to identify the same levels of resolution in the resulting dendrograms. A typical result is shown in figure 4.11. It should be noted, that the first level (that of two clusters) did not occur in all resamples.

It is now straightforward to calculate the value of $\mathcal{M}$ at each level. We present the results as a function of temperature in figure 4.12. The behavior of $\mathcal{M}$ here is pretty much what we would expect, with two trivial maxima, and one at $T = 0.05$ which corresponds to the "correct" solution, of three clusters.



(a) (b)

Figure 4.11: *SPC* results for a typical resample. figure *(a)* shows sizes of the largest clusters as a function of the temperature. Figure *(b)* shows the susceptibility curve.

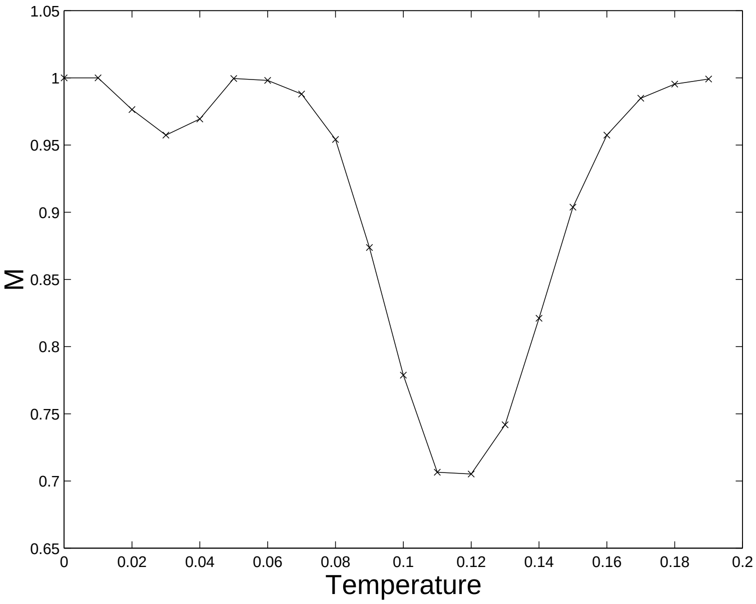


Figure 4.12: $\mathcal{M}$ as a function of the temperature for the three-ring problem.

Chapter 5

Resampling and *SPC* graph connectivity

The input for the *SPC* algorithm is a list of dissimilarity measures of neighboring points. The preprocessing scheme should therefore decide which pairs of points should be considered neighbors, and what is the dissimilarity between those points. This dissimilarity measure can be obtained in many different ways: it may be a distance in a metric space, a correlation function, *etc.* . The choice should be the most natural one for the problem.

The choice of which points are considered neighbors is less natural. Generally, the concept of neighboring points has nothing to do with the problem at hand; it is just a requirement of the algorithm. Moreover, a user of the *SPC* method, who is not interested in the details of the algorithm, is not aware of the meaning of this choice.

We may think of determining which points are neighbors as constructing a graph, in which each vertex is a data point, and each edge connects two points which are neighbors. A useful way of constructing this graph is by using the *K-mutual neighbors* criterion. According to this criterion, two points are considered neighbors if the first point is one of the K closest points to the second point, and vice versa. K is a parameter, which should be supplied externally. The features of this method are briefly described in section 5.2.

The value of K determines the connectivity of the graph, and as turns out, changing the connectivity may yield different clustering results. In other words, the choice among various values of K may be equivalent to the choice between several possible clustering solutions. This subject is discussed in the next section. In another work [17] we describe a scheme to eliminate the external construction of the graph all together.

Here we take a different approach and compare the validity of the different results obtained by using different values of K. Again, we define good clustering to be the one that represents better the underlying distribution of the data. Resampling methods, described in the previous section, are applied, with the hope that again. values of K that maximize stability under resampling, yield clustering that fits the data best.

This chapter demonstrates the use of clustering validity to establish a good working value for an external parameter of a clustering algorithm. Similar approach has been taken in [3] in order to determine the number of clusters in a problem, which is the external parameter c for the c-shell fuzzy clustering algorithm.

5.1 *SPC* results are affected by the connectivity of the graph

The strength of the *SPC* algorithm lies in its ability to use cooperative behavior in order to determine the real correlation between two neighboring points.

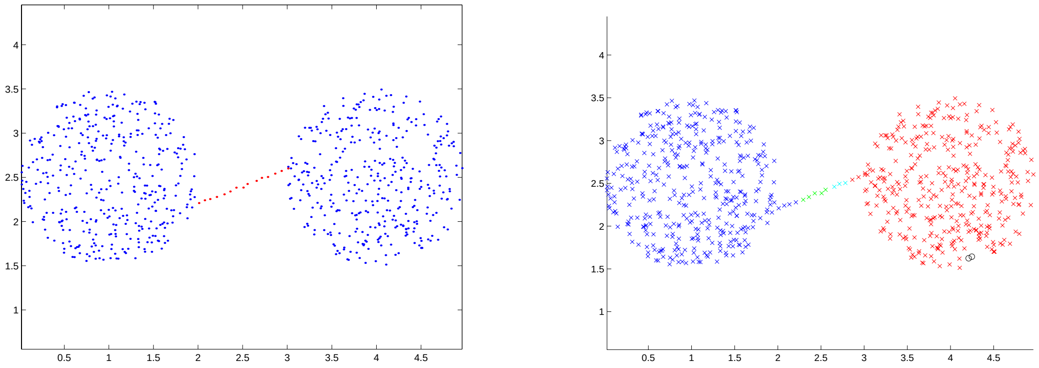
An alternative description for the algorithm can be:

Step 1. For each edge $\langle i, j \rangle$ connecting neighbor points, calculate the correlation function $G_{ij}(T)$ for a range of temperatures.

Step 2. Order the edges according to the temperature T_θ for which $G_{ij}(T_\theta) = \theta$ (one usually takes $\theta = \frac{1}{2}$).

Step 3. Start with a graph where each pair of neighboring points is connected by an edge. To create a dendrogram, remove the edges one at a time, according to the order of **Step 2**.

This description shows the resemblance between the *SPC* algorithm and the agglomerative methods, the main difference being the use of spin-spin correlations rather than distances. The advantage of using correlations rather than distances is demonstrated in figure 5.1.



(a)

(b)

Figure 5.1: The advantage of using correlations rather than distances. We consider the data set shown in *(a)*. Roughly 370 points reside in each cluster. Clearly, the two cluster solution should not be affected by the presence of the filament formed by the 15 points colored in red. Indeed, figure *(b)* shows such a partition found by *SPC* (with K=15). However, if the order in which edges are broken is determined by distances (as it is done in the single-linkage algorithm), then the filament survives long after the clusters start to melt. The partition into two clusters is not observed due to the red filament, which consists of less than 2 percent of the data.

However, if the input graph is very sparse, the correlation between two points is determined mainly by the distance between them. In this case, *SPC* results should be not far from those of the single-linkage agglomerative algorithm, and thus sensitive to noise.

On the other hand, a graph which is too highly connected may blind *SPC* to the cluster structure. To understand this phenomenon, consider the data of figure 5.1(a), either with or without the red points. In a case of very high connectivity, there are many connections both inside each clusters and between the two. At low temperatures the points inside the clusters are highly correlated and, therefore, all connection between them can be considered as a multiple connection between two super-spins. These bonds join forces, and therefore the situation is just as if the distance between the clusters had been much shorter. As the temperature rises, the two clusters start to break, and the bonds connecting them start to act independently. Therefore, the correlation of bonds within the clusters and bonds between them decrease together. There is no temperature at which the two natural clusters are separated while still intact.

We summarize by stating that a graph which is under-connected damages *SPC* 's power to observe collective structure; a graph which is over-connected smears the underlying structure. The level of connectivity must therefore be chosen with some attention. However, once in the correct range, no fine tuning of the edge structure is necessary.

5.2 The K-mutual-neighbor criterion

The task of the K-mutual-neighbors (KMN) method is to select neighbor pairs, or to determine which edges of the similarity/dissimilarity graph to keep. The method can be described in the form of an algorithm:

Step 1. For each point i, order all of its neighbors j by their similarities to i and set k_{ij} to the ordinal number of neighbor j. Note that in general $k_{ij} \neq k_{ji}$.

Step 2. Keep only connections for which $k_{ij} \leq K$ *and* $k_{ji} \leq K$, meaning that point i and point j are at most the K-th neighbors of each other. All other connections are set to zero.

The number of connections that are removed by this method depends on the integer parameter K. A larger K value yields a denser graph, or a higher connectivity.

The main feature of the KMN method is that it separates (*i.e.* removes connections between) regions that have different densities. In order to explain this behavior, we can picture the method, for the case of data residing in a metric space, in the following way. Consider a sphere around each data point that includes its K nearest neighbors. The connection between two points is kept only if both are within the spheres of each other.

In a *uniform* density, it is likely that if point i is inside the sphere of point j then also point j is inside the sphere of point i. Therefore, in the uniform density case, as a first approximation, points that are closer than the average distance to the K-th neighbor will be connected. On the other hand, if there is a *gradient* in the density, on the scale of the average K-th neighbor distance, then spheres around points in the dilute region will include points from the dense region, whereas points in the dense region will have small spheres around them which will not include the points in the dilute region; hence two such points will not be connected. Therefore, the overall effect of the KMN algorithm is to remove the connections between regions of different densities.

5.3 Using resampling to determine interactions range.

The data we examine is constructed from three 2-dimensional ellipse shaped uniform distributions, which are shifted along the x-axis. The smallest distance between the two left distributions is 10 times the mean nearest-neighbor distance, and the distance between the two right ones is 3 times the mean distance. 764 points were sampled from this distribution (Figure 5.2).

Performing *SPC* analysis of this data, using different K values, gives rise to three different clustering solutions. For very low (smaller than 3) and very high (larger than 50) values of K, no macroscopic cluster is identified (except the one which includes all the data).

Using K values between 5 and 20, *SPC* identifies three major clusters, corresponding to the three components of the distribution. At higher tem-

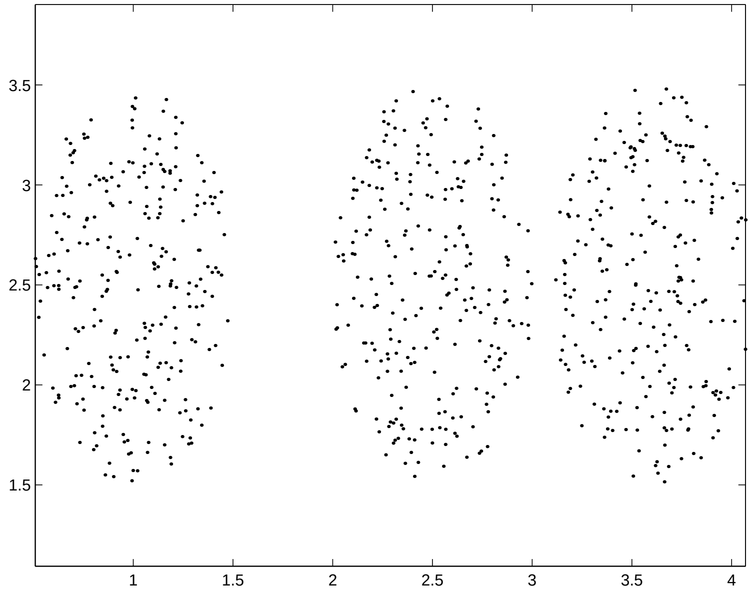


Figure 5.2: 764 points, sampled from a three-component two-dimensional distribution

.

peratures these clusters break up into several sub clusters.

At higher K values (above 20) *SPC* is unable to distinguish between the first two clusters. This is exactly the case when the large number of edges lowers the resolution to a point where the gap between two clusters is unnoticeable. Typical dendrograms are shown in figure 5.3.

We would like to identify the optimal value of K, based on the robustness of the different solutions. This is done by a variant of our resampling scheme. Let us first describe the new scheme, and then comment about why it is needed.

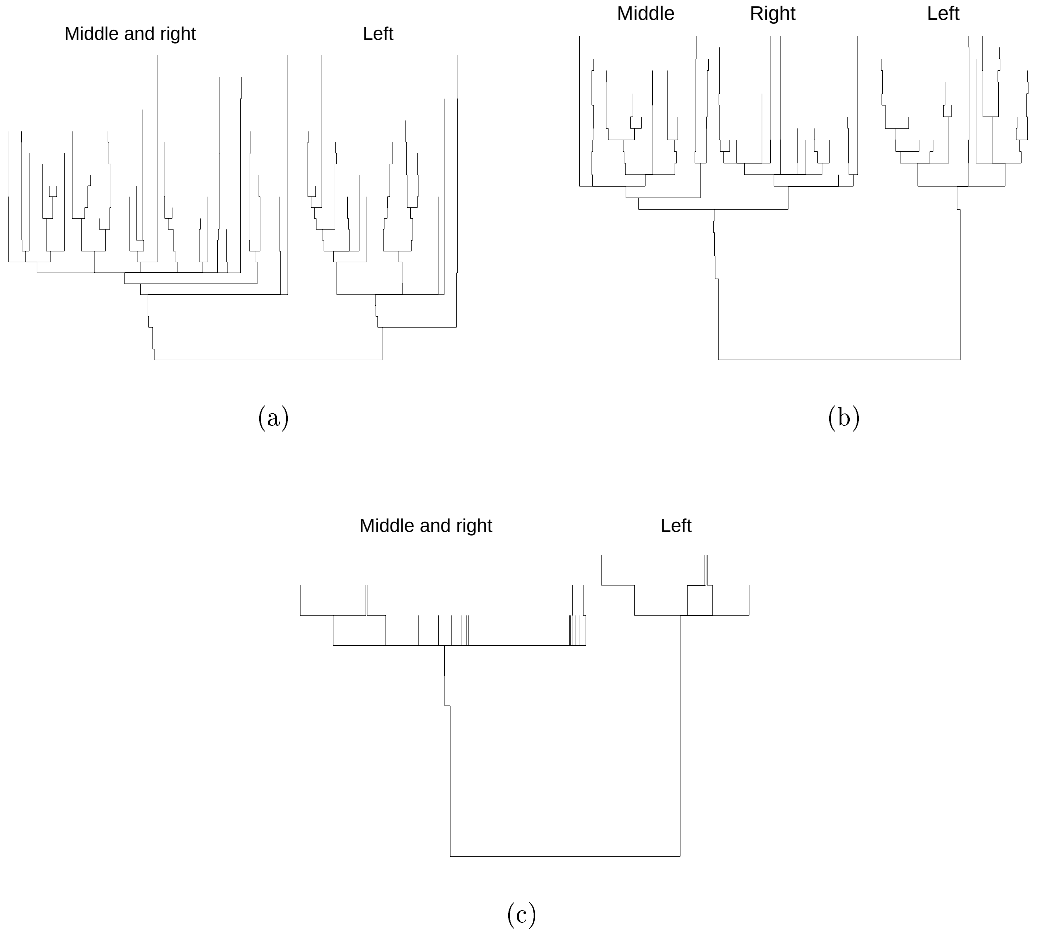
Consider once more a data set of N points. First, we choose a subset A of N_1 points out of the original data. We then divide the remaining $N - N_1$ points to two equal sets, B_1 and B_2. Finally, we construct two data subsets, C_1 and C_2 such that $C_i = A \cup B_i$, $i = 1, 2$.

The *SPC* algorithm is applied to each one of the two data sets C_1 and C_2 separately, yielding a clustering solution for each set. The clustering solution is chosen to be the last level of the dendrogram where all clusters are macroscopic. To compare the two solutions, we look at the data points common to both sets (*i.e.* the subset A), and mark a success for each neighbor pair among them whose members either belongs to the same cluster in both solutions, or belong to different clusters in both solutions. The whole procedure results in a score, which is the number of successes divided by the number of pairs.

The whole process is done with some value of K. Fixing this value, we repeat the whole process m times ($m \simeq 20$ turns to work rather well). The figure of merit for this K value, $\mathcal{M}'(K)$ is just the average of the m scores obtained from the m runs. This score is not very far from our previous score $\mathcal{M}$, except that here we measure the correlation between the clustering assignments obtained for two resamples, rather than between the cluster assignment of a resample and that of the full sample.

This procedure is carried out for all values of K under investigation. A stable solution is characterized by a score close to one.

The subsets used in this method are, of course, smaller than the original set. There is no reason to believe that the optimal K value for sets of different sizes should be the same, and indeed it is not. However, it is reasonable to assume that there is a correspondence between the two. This correspondence should rely heavily on the structure of the data, and thus cannot be determined universally.



(a)

(b)

(c)

Figure 5.3: Clustering solutions for (a) $K = 5$, (b) $K = 10$ and (c) $K = 40$. Branches are identified with the three ellipses. The dendrogram of (b) is the only one to include a level at which the three ellipses are separated.

For this reason it is difficult to calibrate the parameter under investigation a priory as was done before, when discussing the resolution parameter (*e.g.* using $a' = a/f$ in section 4.1). The resampling scheme presented here does not need this calibration, and allows it to be performed after the results of resampling have been obtained.

Figure 5.4 shows the different values of $\mathcal{M}'(k)$ for our two-dimensional three cluster example. Here we used 20 pairs of resamples, each consists of 510 points with a common subset of size $N_1 = 254$. The maximum of this curve at $K \simeq 10$ recovers that clustering solution which indeed identifies the three clusters.

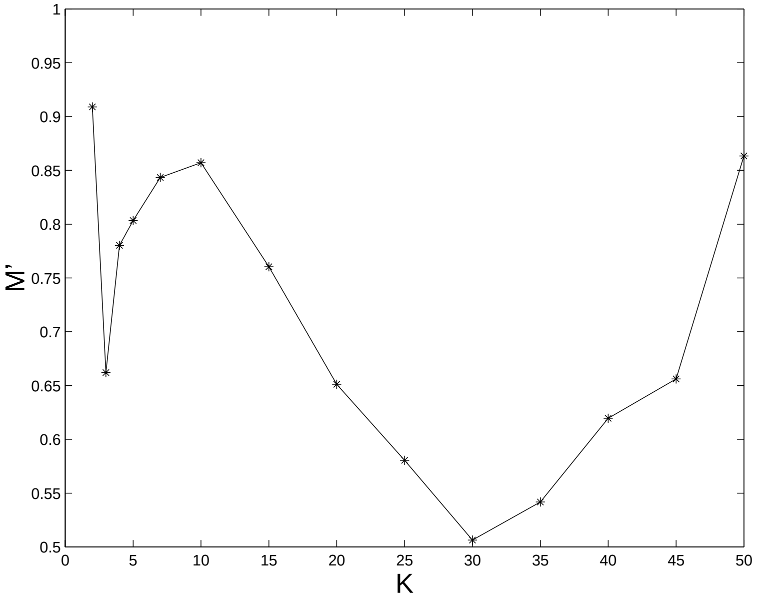


Figure 5.4: The stability score $\mathcal{M}'(K)$ for the two-dimensional three-cluster example. This curve has a peak at $K \simeq 10$, which signals a reliable clustering solution.

In chapter 6 we follow the same method to determine the optimal working value for K for clustering analysis of DNA-chip data.

Chapter 6

Validation method applied to DNA microarray data

6.1 Introduction

The relation between genetic variability and phenotype is a central theme in modern genetics. To understand genetic variation and its consequences on biological function, an enormous effort is made to compare among sequences in various conditions. Conventional nucleic acid sequencing technologies make use of analytical separation techniques to resolve sequence at the single nucleotide level. In this kind of methods, however, the effort required increases linearly with the amount of sequence. In contrast, biological systems read, store, and modify genetic information by molecular recognition. Each DNA strand carries with it the capacity to recognize a uniquely complementary sequence through base pairing. The process of recognition, or hybridization, is therefore highly parallel, as every nucleotide in a large sequence can in principle be queried at the same time. Thus, hybridization can be used to efficiently analyze large amounts of nucleotide sequence.

This idea is the basis for novel microarray-based techniques for high-throughput monitoring of gene expression. The ability of nucleic acids to hybridize with other nucleic acids with a complementary nucleotide sequence lies at the heart of this technology. For example, a piece of DNA with the sequence ATGGCT will readily bind to a DNA with the sequence TACCGA. However, even one nucleotide which is not complementary would reduce significantly the affinity of the binding.

There are two leading methods in the DNA microarray community. One method involves using microarrays of cDNA clones as gene-specific hybridization targets to quantitatively measure expression of the corresponding genes. About 10,000 cDNAs can be robotically spotted onto a microscope slide. A two-color fluorescence labeling and detection scheme facilitated sensitive differential expression analysis of different tissues [20]. This enables the evaluating the response to external perturbation (such as heat shock [21]), the comparison of different phases of a time-sequence (*e.g.* cell division cycle [22], response to radiation), or comparison between different subjects.

A second method uses oligonucleotides probes, designed to match specific sequences. In this way the most informative subset of probes is used [23]. Manufacturing of the array is based on photolithographic masking techniques, used in the semiconductors industry. Using this technique, high density arrays with a precise geometric pattern of hundreds of thousands of short pieces of DNA with defined sequences is built up on a small glass chip. A single chip can analyze in parallel up to 10,000 different sequences. Measurements can either be comparative (using a two-color scheme) or absolute.

DNA microarrays provide a simple and natural tool for exploring the genome, in a systematic and comprehensive way. Several features make it exceptionally well suited for exploratory research [24]: it is universal, fast, fully automated, and relatively cheap. But its main feature is the large scale at which it enables instantaneous look at the activity in a cell.

The main challenge today is no longer in the expression arrays themselves, but in developing experimental designs to exploit the full power of a global perspective. The greatest challenge is analytical [25]. While early expression profiling experiments involved comparing just two samples, experiments today involve tens of different tissues. One expect to gain deeper biological insight from examining datasets with scores of samples – for example, multiple time points from multiple cell lines treated independently, or tissues coming from different patients of several types of disease. The result is an enormous amount of data, from which biological as well as clinical implications should be drawn.

In these cases, each gene defines a point in D-dimensional space (where D is the number of samples studied), and functional similarities are likely to reveal themselves as 'clusters' in this space. Several authors suggested different clustering methods for the task of revealing meaningful clusters

[22, 26, 27]. The question of the similarity measures, however, has not been fully investigated yet.

The main problem is still evaluating the results of these methods. In particular, a method to identify groups of genes which are reliable, and may be candidates for a deeper analysis, is still absent. We suggest using clustering validation methods, suggested at the previous chapter, for these purposes.

Other informatics issues are related to DNA microarrays, including data warehousing, integration with biological databases, and visualization [28].

6.2 Colon Cancer experiment

In this section we describe clustering analysis of oligonucleotide array data, that consists of 2000 gene expression profiles for 62 tissues, 40 of which are colon cancer tumor and 22 are normal. This data was originally analyzed by Alon *et al.* [27]. Using a variant of the deterministic annealing method [19], they performed cluster analysis on the tissues and found two large clusters that coincide with independently determined classification into tumor and normal tissues.

As mentioned at the beginning of this chapter, developing analysis methods for DNA microarray data is of fundamental importance. We therefore revisited this experiment, and suggested a systematic method to zoom in on clusters of genes which cooperatively differentiate tumor tissues from normal ones. Full account of this work is give elsewhere [29].

In this section, we demonstrate the use of our resampling cluster validity method for this DNA microarray data. After a brief account of the clustering analysis, we present the application of resampling methods for clustering tissues as well as genes. We then compare of these results with known tagging of the genes.

6.2.1 The data set

The data set is composed of 40 colon tumor samples and 22 normal colon tissue samples, analyzed with an Affymetrix oligonucleotide array complementary to more than 6500 human genes and ESTs (expressed sequence tags). The genes are represented on a set of 4 chips. Following the analysis performed by Alon *et al.* [27], we use only the 2000 genes of highest intensity[1]. The expression data can be described as a matrix $\mathcal{A}_{ij}$, where $i = 1, \ldots, 2000$ is the gene index and $j = 1, \ldots, 62$ is the tissue index. This data is available on the web at http://wwww.molbio.princeton.edu/colondata/.

To analyze the genes, we view each row of $\mathcal{A}_{ij}$ as a vector in a 62 dimensional metric space. The data to be clustered consists of 2000 such points. Since in most cases [27, 30, 22], one is interested in identifying and clustering together genes that have correlated expressions, the Pearson correlation

[1] We sort the genes according to their minimal expression over the tissues, and include in our data the highest 2000.

coefficient is used as a similarity measure between the genes. Equivalently, one can normalize each row of the matrix, such that its mean vanishes and its L_2 norm is unity. The Euclidean distance between two normalized genes, α and β, is related to the Pearson correlation by

$$d^2_{\alpha\beta} = 2(1 - \mathrm{Corr}(\alpha, \beta)). \tag{6.1}$$

The normalized data constitutes a new matrix, $\mathcal{G}$, where

$$\sum_j \mathcal{G}_{ij} = 0 \ , \ \sum_j \mathcal{G}^2_{ij} = 1 \quad \forall i. \tag{6.2}$$

Another way to view the data is to consider the tissues as our (62) data points in a 2000 dimensional space. This will allow us to cluster the tissues on the basis of their expression profile over the entire set of genes. For this view we construct a new matrix $\mathcal{T}$, obtained by normalizing the *columns* of the matrix $\mathcal{A}$. The elements of $\mathcal{T}$ are

$$\mathcal{T}_{ij} = \frac{\mathcal{A}_{ji} - \sum_k \mathcal{A}_{ki}/n}{\sqrt{n-1}\mathrm{Std}_k \mathcal{A}_{ki}} \ . \tag{6.3}$$

Each of the 62 rows of $\mathcal{T}$ represents the gene expressions of a different tissue, yielding 62 data points in a 2000 dimensional space.

6.2.2 Clustering the tissues

The main purpose of clustering the tissues is to check whether the tumor and normal tissues can be naturally separated on the basis of gene expression data. If this is indeed possible, one can proceed to the next step and identify the genes that are responsible for the separation.

We used *SPC* to cluster the tissues [29], and identified two clusters of tumor and normal tissues with high accuracy and efficiency. We chose the value $K = 12$, using a method which may be considered supervised; expecting to find two classes, we settled for a value of K which has 'got the job done'.

In general, we want the clustering procedure to be unsupervised all the way. This is of course the case when no prior knowledge of the data is available. Here we use the resampling mechanism to identify a good K value.

Different solutions

Using different values for the parameter K yields different clustering solutions. These solutions can be seen in the dendrograms of figure 6.1.

When using *SPC* with $K = 5$, the data breaks into the 'correct' two clusters, but then almost immediately it breaks again into five sub-clusters. Looking at the temperature scale, one may argue that the five cluster solution is the stable solution (in terms of section A.2), and one may question the validity of the two-cluster solution.

For $K = 15$, the only break into macroscopic clusters is the one which identifies the two 'correct' clusters. We mark these clusters by arrows in the dendrogram (figure 6.1(b)). The corresponding dendrogram shows that the larger cluster of the two is less dense, and so it 'melts away' while the smaller cluster survives for a wide range of temperatures.

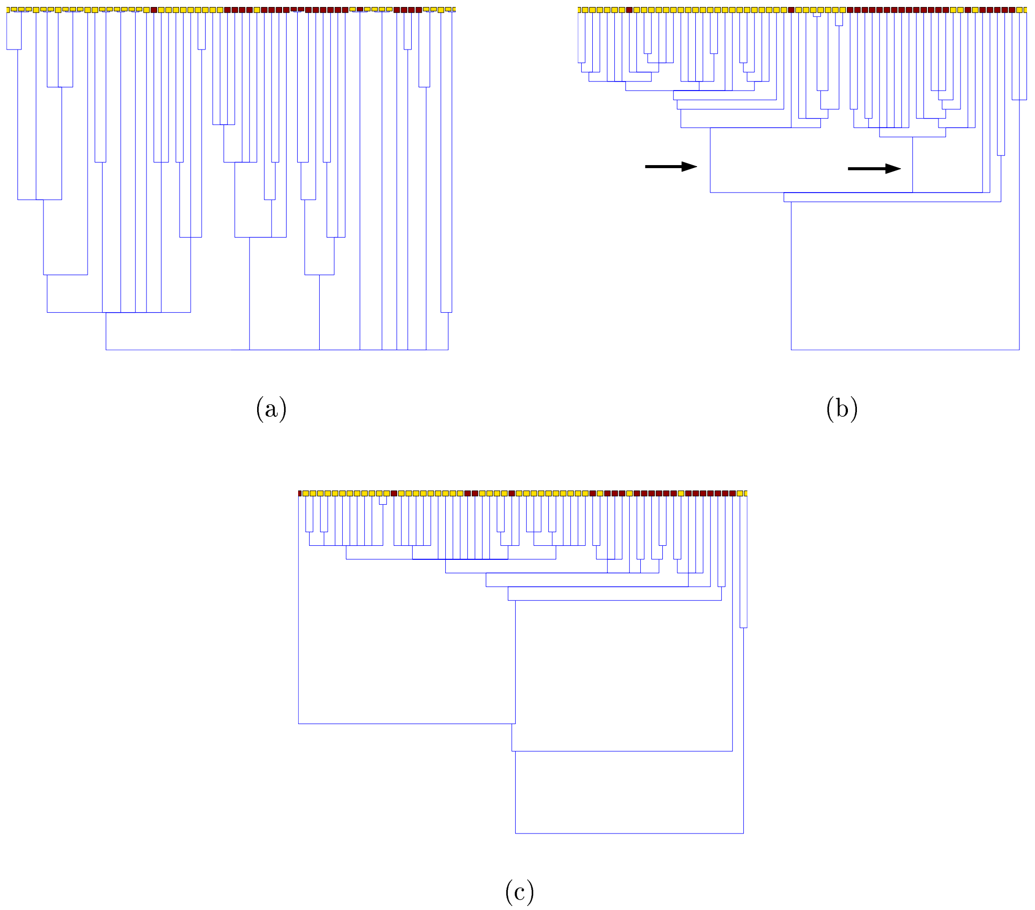
Finally, at $K = 25$ one identifies the separation of the small cluster from the rest of the data. However, the rest of the data breaks immediately into many clusters of order one, so it should be considered as a background rather than another cluster.

Geometric interpretation

The above results suggest the following geometric interpretation, visualized in figure 6.2. We imagine that our data set is constructed from three clusters, where the first (A) is denser than the other two, while those (B and C) are closer to each other then to A. It may very well be that the separation between clusters B and C is a sample artifact.

At low K value, the three clusters are very loosely connected. As the temperature rises, the three break up almost together. This is a manifestation of the fact that small values of K increase the sensitivity of *SPC* to noise (such as the small crack in the data, separating clusters B and C).

Increasing the value of K is manifested more significantly in the number of connection between clusters B and C then in the connections between those clusters and cluster A. Therefore, the global interaction to cluster A gets relatively weaker, and the two major clusters, A and B+C, are separated at some temperature. However, it may be that the large number of connections between clusters B and C makes the interaction between them so strong, that they are not separated anymore, but melt together. This meltdown occurs



(a)

(b)

(c)

Figure 6.1: Clustering solutions for (a) $K = 5$, (b) $K = 15$ and (c) $K = 25$.

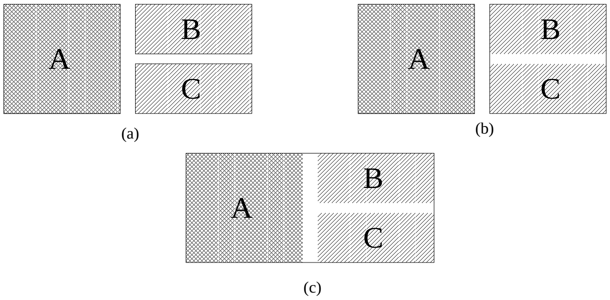


Figure 6.2: Geometric interpretation for different clustering results, for (a) low K values, (b) moderate values, and (c) high values.

while cluster A stays intact, due to its higher density.

As K gets even larger, the whole data set is highly connected, and does not break into large clusters. It just "melts" gradually, starting with the lower density areas (formerly known as clusters B and C), and moving at higher temperatures to the higher density area A.

Resampling method

Let us now go back to our resampling method for validating a clustering solution of the tissues. The question of which value of K to choose may be translated to one on cluster validity. If the 5 cluster structure is a reliable one, a small K value should be selected. If no cluster structure exists in the data, a large K should be chosen, etc.

In order to calculate our figure of merit $\mathcal{M}$, we break the data 20 times into two parts, each of size 2/3 of the data. One third of the points reside in both parts, and their clustering assignment is used to calculate $\mathcal{M}$. For each application of the *SPC* algorithm, we regard the first break into macroscopic clusters as the clustering solution.

The results obtained for several values of K are plotted in figure 6.3. As expected, very low and very high values of K give similarly stable solutions. These are just the cases where the first break in the data already breaks it up to single point clusters. However, At $K = 8$ we observe another peak in M, which corresponds to a break into the two 'correct' clusters. This solution appears also for higher values of K, but in these cases it is not stable to resampling.

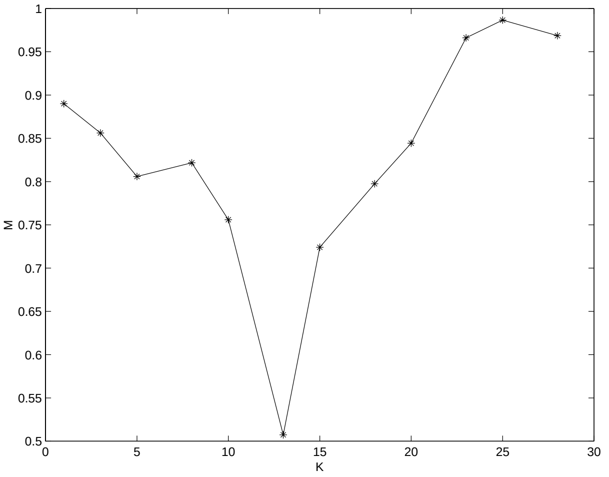


Figure 6.3: Figure of merit $\mathcal{M}$ as a function of K for the colon tissue data.

It should be noted, however, that this optimal value of $K = 8$ correspond to a data set which is smaller than the original one. In order to find the corresponding K values for the full sample, we check for which K are the clustering assignments of the full sample most similar to the assignments obtained for the resampled subsets. We find that when using $K = 10$ for clustering the full sample we obtain the most similar results[2] to clustering the resampled subsets with $K = 8$.

We conclude that using $K = 10$ for the full sample is optimal for our case. This value is in agreement with our understanding that this value should be low (in order to be sensitive to the substructure of the data), but not too low (in a way that would not enable to reveal the hierarchy of the structure).

6.2.3 Clustering genes

Clustering the genes serves two purposes. One is to identify groups of genes which act cooperatively. If such cooperative behavior is consistent with the different classes of tissues, one can hypothesize about the role this group of genes serves in the colon cancer.

[2] The similarity between these clustering solutions is measured as defined by the original $\mathcal{M}$.

The other purpose is more exploratory in nature. Having identied cluster of genes, one may ask what makes their expression correlated. Trying to answer this question, one may, for example, identify common promoters of these genes, or alternative classifications of the tissues.

Clustering analysis is therefore applied for the data matrix $\mathcal{G}$, which has 2000 rows corresponding to the 2000 genes, and 62 columns corresponding to the tissues. In [29] we propose a systematic method to analyze the clustering results, and to identify such clusters. Here we take a different approach, and try to identify gene clusters worth inspection based on their reliability, in terms of stability against resampling.

The *SPC* results are shown as a dendrogram in figure 6.4. This dendrogram is the result of clustering the full 2000 genes.

Our resampling scheme has been applied to this data, using 25 resamples of size 1200. This time, however, we calculated $\mathcal{M}$ for each cluster of the full original sample separately. Given a cluster C, we calculate $\mathcal{M}$ only for the points of this cluster, at the temperature at which this cluster is identified. We therefore get a stability merit for each cluster. In figure 6.4 we paint each box, representing clusters, according to this score.

We now focus our attention on clusters of highest scores. First, we take the top 20 clusters. If one of these clusters is a descendent of another one from the list of 20, we discard it. We are left with 6 clusters, which are circled and numbered in figure 6.4 (the numbers are not related to the stability score).

We are now ready to interpret those clusters. The first three clusters consist of known families of genes. Cluster #1 is the Ribosomal proteins cluster. The genes of cluster #2 are all Cytochrome C genes, which are related with energy transfer. Most of the genes of cluster #3 belong to the HLA-2 family, which are histocompatability antigens.

Cluster #4 contains a variety of genes. Some of these genes are related to metabolism. When trying to cluster the tissues based on these genes alone, we find a stable partition to two clusters, which is not consistent with the tumor/normal labeling. At this point, we are still unable to explain this partition.

Clusters #5 and #6 contain also genes of various types. The genes of these clusters have most typical behavior: all the genes of cluster #5 are highly expressed in the normal tissues but not in the tumor ones; And all genes of cluster #6 are the other way around.

To summarize, clustering stability score based on resampling enabled us

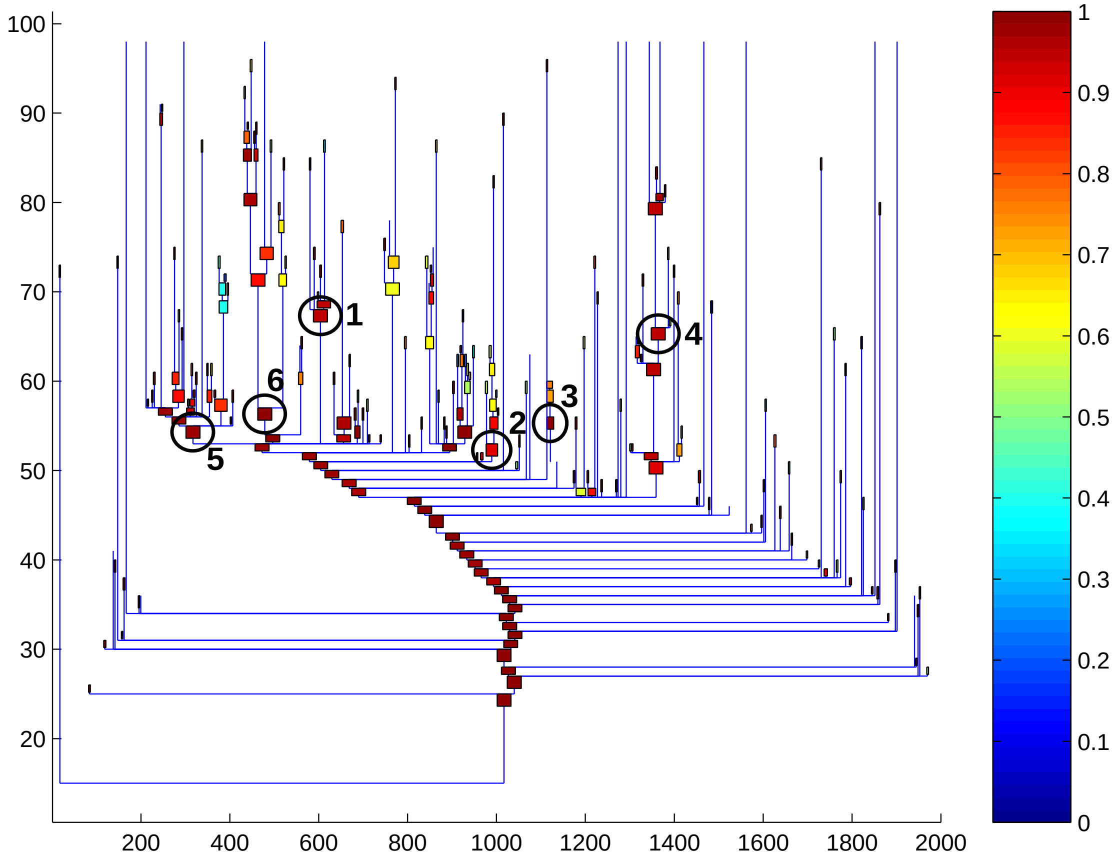


Figure 6.4: Dendrogram of genes. Clusters of size 7 or larger are shown as boxes. Color code is based on $\mathcal{M}$. Selected clusters are circled and numbered, as explained in the text.

to zero in on clusters with typical behavior, which may have biological meaning. Using resampling enabled us to select clusters without making any new assumption, which is a major advantage in exploratory research. The downside of this method, however, is it's computational burden. In this experiment we had to perform clustering analysis 20 times for a rather large data set. This would be the typical case for DNA microarray data.

Chapter 7

Summary

This work proposes a method to validate clustering analysis results, based on resampling. It is assumed that a cluster which is robust to resampling is less likely to be the result of a sample artifact or fluctuations.

The strength of this method is that it requires no additional assumptions. Specifically, no assumption is made either about the structure of the data, the expected clusters, or the noise in the data.

We introduced a figure of merit, $\mathcal{M}$, which reflects the stability of the cluster partition against resampling. The typical behavior of this figure of merit as a function of the resolution parameter allows clear identification of natural resolution scales in the problem.

The question of a natural resolution levels is inherent to the clustering problem, and thus emerges in any clustering scheme. The resampling method introduced here is general, and applicable to any kind of data set, and to any clustering algorithm.

Using a simple one-dimensional model, we got an analytical expression for our figure of merit, which enabled us to determine its behavior as a function of the resolution parameter. Local maxima are identified for values of the parameter corresponding to stable clustering solutions. Such solutions can either be trivial (at very low and very high resolution); or non-trivial, revealing internal structure of the data.

We tested this behavior numerically on several data sets, of different levels of complexity: a one dimensional data set, very similar to the one considered in our model; a two-dimensional toy problem, where we used *SPC* as our clustering algorithm; and real life DNA microarray data. In the first two cases local maxima, corresponding to correct solutions, were observed. In the last case, relatively high values of the figure of merit enabled us to identify genuine clusters of genes.

The figure of merit $\mathcal{M}$ was also used to determine an optimal value for an important external parameter of the *SPC* algorithm, namely the connectivity level of the input graph. At the optimal value the clustering solution is stable against resampling and, therefore, the figure of merit $\mathcal{M}$ has a local maximum. We demonstrated this application of the figure of merit for two-dimensional toy data and for tissue clustering, based on DNA microarray data.

Our resampling scheme enable us to identify the most stable clustering solution among several alternatives. It does not, however, assign a statistical significance to this solution. This should be the subject for further research.

The fast developing area of DNA microarrays calls for the formulation of an analysis scheme, which incorporates not only good similarity measures and clustering methods, but also methods to validate the results. We have shown here that resampling techniques, together with a validity index which is free from uncontrolled assumptions, may serve well in this area. This also calls for more research.

Appendix A

Cluster validity indices

The most common method of clustering validation is the use of validity indicators. Given a clustering solution, the calculated value of an indicator should reflect, in some sense, how well this solution meets a certain criterion. This criterion incorporates the the features one expects a good clustering solution to have.

Most cluster validity indicators are based on some geometrical interpretation of the notion of clustering. Let us first present a few examples.

The first index [2] assumes that the clusters are compact. It then states that a criterion for a good partition of the data into subgroups should fill the following requirements:

1. Clear separation between the resulting clusters.
2. Minimal volume of the clusters.
3. Maximal number of data points concentrated in the vicinity of the cluster centroid.

(A.1)

Given a clustering solution, an index W is calculated, to measure how well this solution meets these requirements. Let c_i be the i'th cluster with $\vec{g}_i$ its centroid. Following [2], we define the within cluster dispersion

$$S_i = \sqrt{\frac{1}{|c_i|} \sum_{\vec{x}_j \in c_i} \left(\vec{x}_j - \vec{g}_i\right)^2} \ , \tag{A.2}$$

and the between cluster separation

$$T_{ij} = |\vec{g}_i - \vec{g}_j| \ . \tag{A.3}$$

The index is then given by

$$W = \frac{1}{C} \sum_{i=1}^{C} \max_{j \neq i} \frac{S_i + S_j}{T_{ij}} , \tag{A.4}$$

where C is the number of clusters.

For each pair of clusters, the ratio between the mean within cluster dispersion and the between cluster dispersion is measured. In order to calculate W, the worst counterpart for each cluster, *i.e.* the cluster for which this ratio is maximal, is taken. Therefore, a smaller value of W reflects a clustering solution which better fulfills the requirements.

There are two major assumptions involved in the requirements A.1. First, one assumes that the data points reside in a metric space, so that one can identify the centroid of a cluster, and calculate the distances of points from it. Next, it is assumed not only that clusters are compact, but also that they are spherical about a centroid. This assumption is correct for a very small subset of clearly clustered data sets. For example, the data set of figure 4.9, as well as the one of figure A.1, do not share this property.

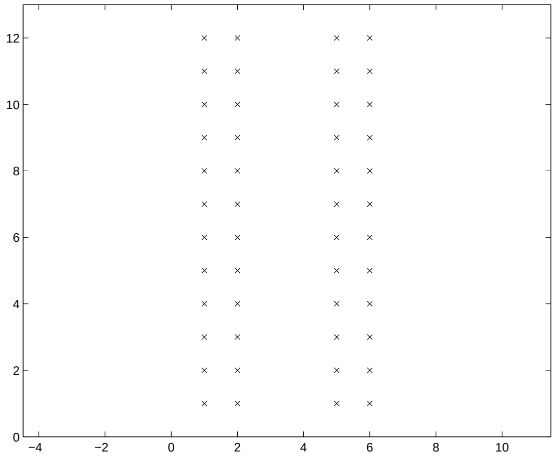


Figure A.1: Example data set. 48 points reside in two non-compact clusters.

However, many clustering algorithms (*e.g.* DA, EM, c-shell [3] and more) incorporate precisely theses assumptions into their search process. It is therefore not surprising that this clustering validity index, and many similar to it, are widely used to compare between these methods.

The second index [1] uses more local notions. It is based on the observation, that in order to include a point in a cluster, one expect this point to be closer to the points of this cluster than to other points. The family of indices which try to reflect this feature are all based on comparing between distances among points within a cluster, to distance between points of different clusters.

One of the members of this family, perhaps the most straightforward one, is the index $\tilde{R}$, given by

$$\tilde{R} = \frac{1}{C}\sqrt{\sum_c \left(\frac{\tilde{W}_c}{\tilde{M}_c}\right)^2} \ , \tag{A.5}$$

with

$$\tilde{W}_c = \frac{1}{n_c^2}\sum_{i,j\in c} d_{ij} \ , \tag{A.6}$$

and

$$\tilde{M}_c = \min_{c'} \frac{1}{n_c n_{c'}} \sum_{i\in c, j\in c'} d_{ij} \ . \tag{A.7}$$

Here c is the cluster index, $c = 1, \ldots, C$; c' is the cluster which is 'closest' to the cluster c; n_c and $n_{c'}$ are the cluster sizes. Clustering assignment which reduces the value of $\tilde{R}$ is considered better.

Again, there is an assumption of compactness both in this index, and in the clustering notion behind it. In any cluster, there may be some points which are far from each other. For example, imagine a spherical cluster, and consider two points which lie on the perimeter, separated by the full sphere diameter. These points reside in the same cluster due to the fact, that they are both close to the cluster core. The distance between these two points contributes to $\tilde{W}$. However, if the cluster is compact, there are not that many such pairs, and the sum in $\tilde{W}$ is not affected strongly by them.

If the clusters are not compact, however, these pairs dominate the sum. For example we refer the reader again to the sample of figure A.1. Two clusters of 60 points, arranged on a cubic lattice with nearest-neighbor distance $a = 1$, are separated along the x axis by a distance $b = 5$. Here

$$\tilde{W}_{c_1} = \tilde{W}_{c_2} = 4.27 \quad ; \quad \tilde{M} = 6.0 \tag{A.8}$$

and

$$\tilde{R} = 0.5 \ . \tag{A.9}$$

By imposing a cluster solution which is compact, even though unnatural to this problem, we can reduce this value of R. Let us break the same data set into two clusters, by breaking it middle way along the y axis. Now we have

$$\tilde{W}_{c_1} = \tilde{W}_{c_2} = 3.53 \quad ; \quad \tilde{M} = 6.72 \tag{A.10}$$

and

$$\tilde{R} = 0.28 \; . \tag{A.11}$$

As expected, the compact cluster solution, though incorrect, gives a better result according to this definition of the index $\tilde{R}$.

A.1 Eliminating compactness assumption by using pairs of neighbors

We would like to introduce now a modified version for this index. In the new version, which we call R, only pairs of neighbor points are considered. Furthermore, we use a rapidly decreasing function of the distance, J_{ij} of equation 2.5, instead of the distance itself. We then have

$$R(T) = \frac{1}{C}\sqrt{\sum_c \left(\frac{W_c}{M_c}\right)^2} \;\; , \tag{A.12}$$

with

$$W_c = \frac{1}{e_c} \sum_{\substack{\langle i,j \rangle \\ i,j \in c}} J_{ij}, \tag{A.13}$$

$$M_c = \frac{1}{e'_c} \sum_{\substack{\langle i,j \rangle \\ i \in c, j \notin c}} J_{ij}. \tag{A.14}$$

Where e_c is the number of edges connecting points of cluster c, and e'_c is the number of edges connecting point of cluster c and points outside it. Note that by moving from the distance d_{ij} to the interaction J_{ij}, we expect now R to be larger for a better solution.

Let us look back at the example of figure A.1. Neighboring points were decided according to the K-nearest-neighbor criterion, with $K = 4$. This time we have

$$W_{c_1} = W_{c_2} = 0.7 \quad ; \quad M = 0.46 \quad ; \quad R = 2.4 \tag{A.15}$$

for the true clusters; and

$$W_{c_1} = W_{c_2} = 0.7 \quad ; \quad M = 0.68 \quad ; \quad R = 1.08 \tag{A.16}$$

for the compact partition. This time, a better solution is identified by the larger R.

Imagine that an analyst has a data set he wants to cluster, and he finds in a book two clustering algorithms, which we call A_1 and A_2. The analyst applied the two algorithms to his data set, and obtains two distinct clustering solutions. Having read this text, he calculates the R index, and gets the results R_1 and R_2, respectively, where $R_1 > R_2$. But what is the statistical significance of this result?

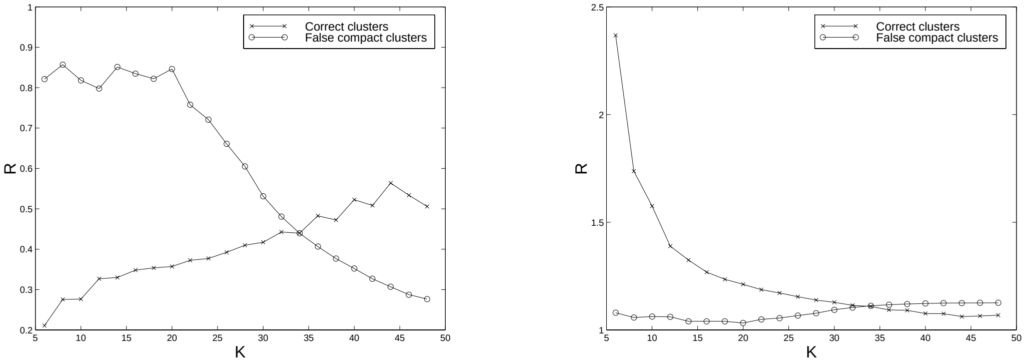
In order to give this kind of results a statistical significance, one should obtain the statistics of the index R. This is done using resampling again, in a method which is based on the *bootstrap* approach [6]. Again, a set of resamples is constructed. For each resample, clustering algorithms, say A_1 and A_2, are applied. The index R is then calculated for every clustering solution of every resample. Using the obtained populations of R_1 and R_2 values, the standard deviations σ_1 and σ_2 are obtained.

It is than straightforward to define a measure, similar to the well-known *t-test*,

$$t = \frac{R_1 - R_2}{\left(\sigma_1^{-1} + \sigma_2^{-1}\right)^{-1}}, \tag{A.17}$$

which measures the difference between the two original values of R in terms of its standard deviation. This measure provides a statistical meaningful method to compare the two results.

We note that the value of R is related to the method of determining the neighboring pairs. Specifically, if one chooses to work with the K-nearest-neighbor criterion, R depends on K. This dependence, however, is not too sensitive. In figure A.2 we show the values of R for different values of K.



(a)

(b)

Figure A.2: *(a)* The index R calculated for different sets of neighboring pairs, determined by the K-nearest-neighbor criterion. Remember that a better solution is characterized by a larger value of R. As the number of neighbor pairs gets too large, R prefers compact clusters. *(b)* For the same data, $\tilde{R}$ is calculated using only neighbor points; lower R corresponds to a better solution.

A.2 Temperature range is a measure of stability

All methods which average distances within clusters and between them, have a problem which we call counting irrelevant distances. To explain what we mean by irrelevant distances, let us define what relevant distances are.

There are two distance scales, or resolutions, in the life of a cluster. The first scale is the one which separates this cluster from its surroundings. This may be the width of the valley in the distribution of data points that surrounds it, its distance from the nearest cluster, etc. The second scale, is the one separating its sub-clusters, or the resolution at which it breaks up.

The relevant distances between data points of the cluster are those on either one of the two scales. Obviously, counting all distances (as suggested in $\tilde{R}$) include many distances which have nothing to do with these scales (for example, the distances between points on two different sides of the cluster).

Counting only pairs of neighbor points reduces the number of irrelevant distances that are counted, but does not eliminate them completely. No method exists that can distinguish between pairs of relevant distances and pairs of irrelevant ones.

We present here a method to overcome this problem, which is unique to the *SPC* algorithm. In figure A.3 we plot the correlation functions G_{ij}, as a function of the temperature, for all neighbor pairs in the data of figure A.1. Let us focus our attention on the left cluster. This cluster is formed at temperature $T = 0.037$, where the correlation functions corresponding to the edges connecting it to the world (painted red in the figure) decrease below the threshold value $\theta = 0.5$. It is broken when the correlations corresponding to internal edges (painted green) also decrease below the threshold, at $T \simeq 0.1$.

The distance between the two groups of curves, the one corresponding to correlations between the cluster and the world, and the other corresponding to correlations within the cluster, is a measure for the stability of the cluster. In [17] we argue that the best way to measure the distance between correlations is by subtracting the temperature integrals of the two curves. If one approximates these curves by a unit step function, which changes from one to zero at the temperature where the original function gets the value 0.5, than the difference of the integrals is just the difference of the corresponding temperatures. This difference is then normalized by the mean interaction

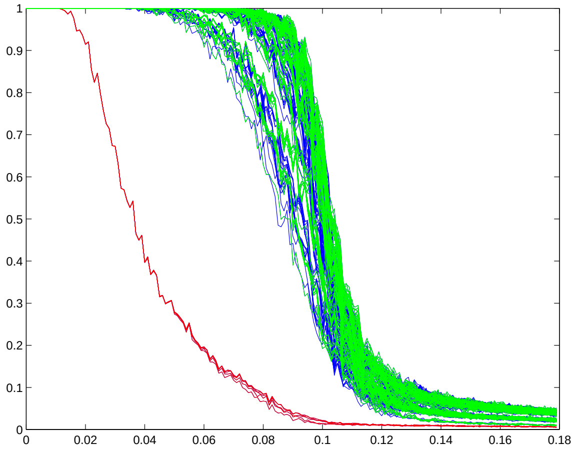


Figure A.3: Correlation functions for neighbor pairs of the data set in figure A.1. Lines in red correspond to edges connecting the two clusters, lines in green correspond to internal edges of the left cluster.

strength, to obtain the dimensionless stability index

$$\mathcal{Q} = \frac{T_{\text{int}} - T_{\text{ext}}}{\bar{J}} \,. \tag{A.18}$$

Here T_{ext} is the defined by $G(T_{\text{ext}}) = \frac{1}{2}$ for the edges of the group which connects the cluster to the outside world[1] and similarly $G(T_{\text{int}}) = \frac{1}{2}$ for edges which connect its sub-clusters.

It is intuitively appealing to claim that clusters which survive for a wider range of temperatures are more reliable. This definition of $\mathcal{Q}$ provides a geometrical interpretation of this observation. Clusters which survive for wider ranges of temperatures are those for which the distance scale which separates the cluster from its surrounding is significantly different from the one which breaks it to pieces.

Obtaining statistics for the index $\mathcal{Q}$ is difficult. Therefore, $\mathcal{Q}$ should be used mainly to order resulting clusters according to their reliability. In then next section we demonstrate the application of this method.

[1] Due to noise, it may be that different edges of this group have slightly different such temperatures. In such case, we choose the highest of these temperatures.

A.3 Yeast cell cycle experiment

The biology literature of recent years presents a lot of evidence for cell cycle-regulated genes. Such regulation may be required for the proper functioning of mechanisms that maintain order during cell division, allow conservation of resources and more.

Many cell cycle-regulated genes are involved in processes that occur only once per cell cycle, including many that controls the cell division. The cell division cycle is thus a self-regulating mechanism.

A series of experiments were cpnducted by Spellman *et al.* [31] in order to identify cell-cycle regulated genes in the gnome of *Saccharomyces cerevisiae* yeast. These experiments utilized cDNA microarrays with cDNAs representing the full yeast genome.

Spellman *et al.* used several different methods to synchronize the yeast cells in the sample before microarray hybridization. We limit our discussion here to one experiment where the synchronyzation was done by adding an α factor to the sample. Samples for hybridiztion were taken every 7 minutes for almost 140 minutes. Details of the experiment can be found in [31]. The data is available at URL http://cellcycle-www.stanford.edu.

Following Eisen *et al.* [22], we use in our analysis only the 2467 genes whose functional annotation is known. The data from this experiment is thus a 2467×18 matrix, $E_{ij}, \quad i = 1, ...2467$ and $j = 1, ...18$.

A.3.1 Clustering genes with *SPC*

In order to find groups of cell cycle-regulated genes, or other groups of genes with potentially interesting behavior, we perform clustering for the genes. We therefore regard the genes as 2467 data points in a 18 dimensional space.

The coordinates of each gene were normalized in the standard way;

$$Y_{ij} = \frac{E_{ij} - < E_i >}{\sigma_i} \tag{A.19}$$

where

$$< E_i > = \frac{1}{18} \sum_{j=1}^{18} E_{ij} \quad ; \quad \sigma_i^2 = \frac{1}{18} \sum_{j=1}^{18} E_{ij}^2 - < E_i >^2 \quad . \tag{A.20}$$

Euclidean distance between the rows of the matrix Y_{ij} is proportional to the Pearson correlation, commonly used in the DNA microarray community, between rows of the original matrix E_{ij}.

The data were clustered using the *SPC* algorithm. The neighboring points were decided according to the *K-mutual-neighbor* criterion, described in chapter 2, with $K = 10$.

The resulting dendrogram is shown in Fig A.4. Each node represents a cluster at some level. The boxes are clusters whose size exceeds 6.

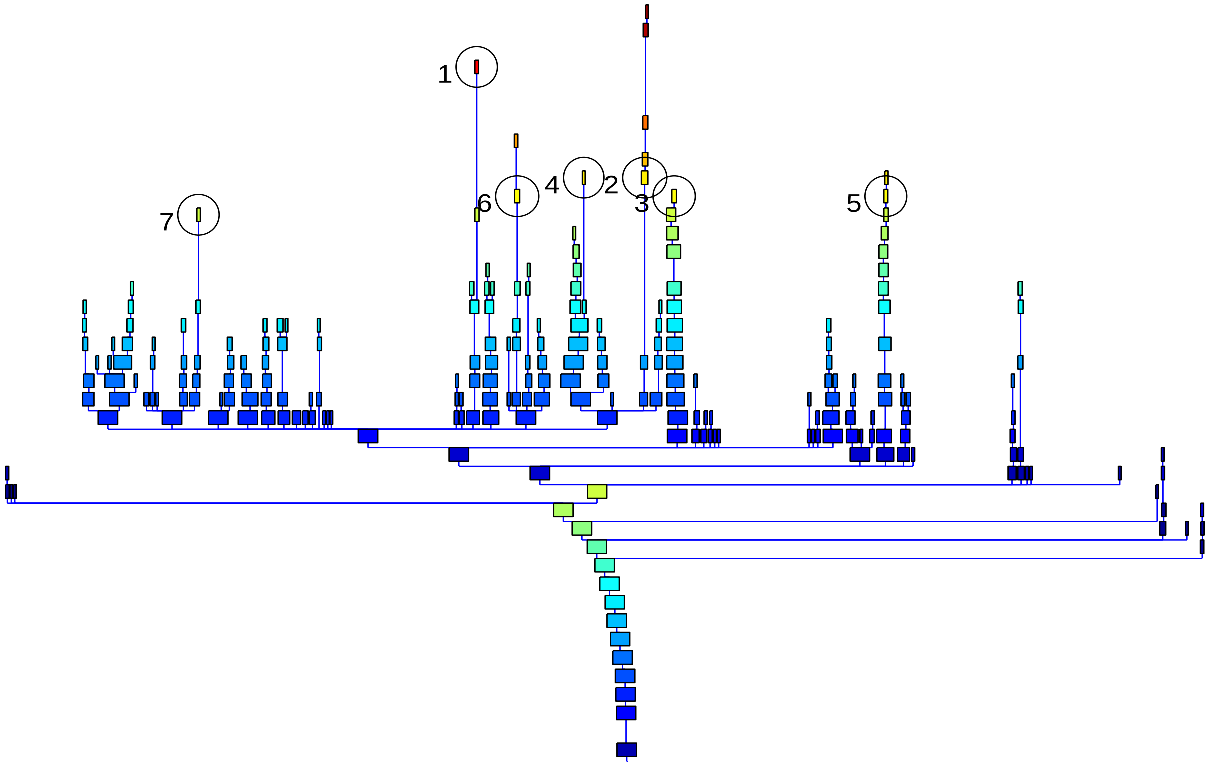


Figure A.4: Dendrogram of genes, according to *SPC* . Boxes are clusters containing 7 genes or more. Coloring is according to the $\mathcal{Q}$ score. Circled clusters are the selected ones, as explained in the text.

The branches of the resulting dendrogram were ordered in the following way: say at some point a cluster C has split into C_1 and C_2. Say of these two C_1 splits first, into $C_1^{(1)}$ and $C_1^{(2)}$. These two sibling branches are ordered according to the proximity of their centroids to the centroid of their "uncle", C_2. The ordering obtained from this procedure is believed to reflect the structure of the data.

A.3.2 Cluster validity index is used to identify interesting clusters

In section A.2 we described the index $\mathcal{Q}$, which identifies stable clusters as those which survive over a long range of temperatures. We claim that the correlation between points residing inside such a cluster is significantly larger than the highest correlation between this cluster and its surroundings.

In terms of gene expression, this argument is translated to the argument, that the time variation of genes residing in such a cluster is significantly correlated among themselves. If the only synchronized process in the different cells is the cell division cycle, then one expects such genes to be cell cycle-regulated.

We calculated the score $\mathcal{Q}$ for each one of the clusters which appear in the dendrogram of figure A.4. The score of each cluster determines its color in this figure. Histogram of the scores is shown in figure A.5.

The score $\mathcal{Q}$ can now be thresholded in order to select interesting clusters. We select a cluster if its $\mathcal{Q}$ score is higher than 0.9, while its parent's score is not. This way we do not bother ourselves with sub clusters of selected clusters. The selected clusters are the ones which are circled in figure A.4. The mean time profile of these clusters is shown in figure A.6.

We would like now to interpret the selected clusters. By inspecting the mean profiles, it is evident that clusters 1, 2 and 3 are cell cycle-regulated. Indeed, we identify in these clusters many genes which are known to be cell cycle-regulated.

Like all living cells, the yeast cell cycle consists of four phases:
G1→S→G2→M→G1 . . . , where S is the phase of DNA synthesis, M stands for mitosis, and G1 and G2 are the gap phases. In cluster #1 we find genes which are active at the G2/M-phase of the cell cycle; Cluster #2 is dominated by histones, which are S phase genes; and cluster #3 contains mostly late

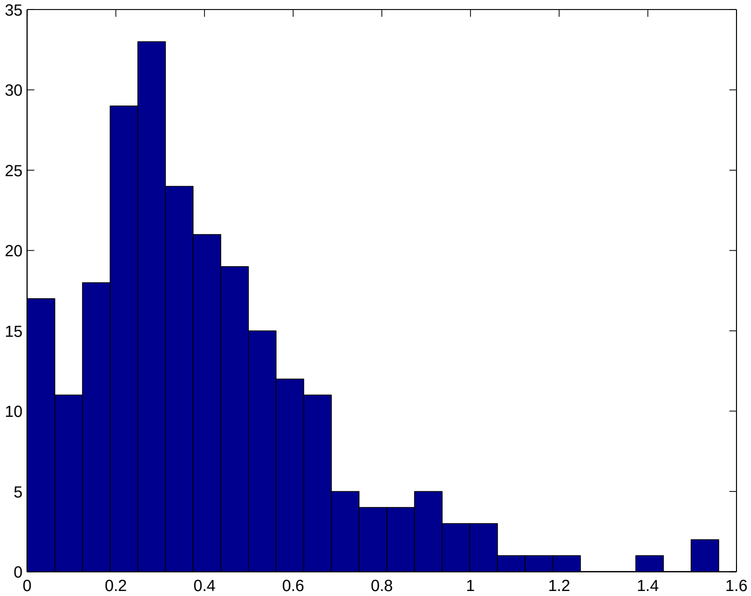


Figure A.5: Histogram of the $\mathcal{Q}$-score for the clusters of figure A.4.

G1 specific genes.

Clusters 4 and 5 are not cell cycle-regulated, as we can see from their mean expression profile. This claim is supported by the fact, that in these cluster we could not find any gene which is known to be cell-cycle regulated. Nevertheless, the two clusters do have some unique profile. In cluster #4, we clearly identify a transient behavior. This behavior may be related to the fact that the cells were starved during the α factor arrest. Genes of cluster #5 are correlated due to the sharp peak at the second time step. This may be some artifact of the experiment.

Finally, clusters 6 and 7 show more erratic behavior. From the small error bars, we see how well the genes of these clusters are correlated among themselves. It turns out that the genes of these two clusters are are all ribosomal proteins, which we find in all our DNA chip works to be highly correlated.

In another work [32] we have used other methods to identify interesting clusters. We found 11 such clusters. Out of these clusters, 7 clusters share the same branch as the clusters selected here. The other 4 clusters have larger standard deviation from the mean profile.

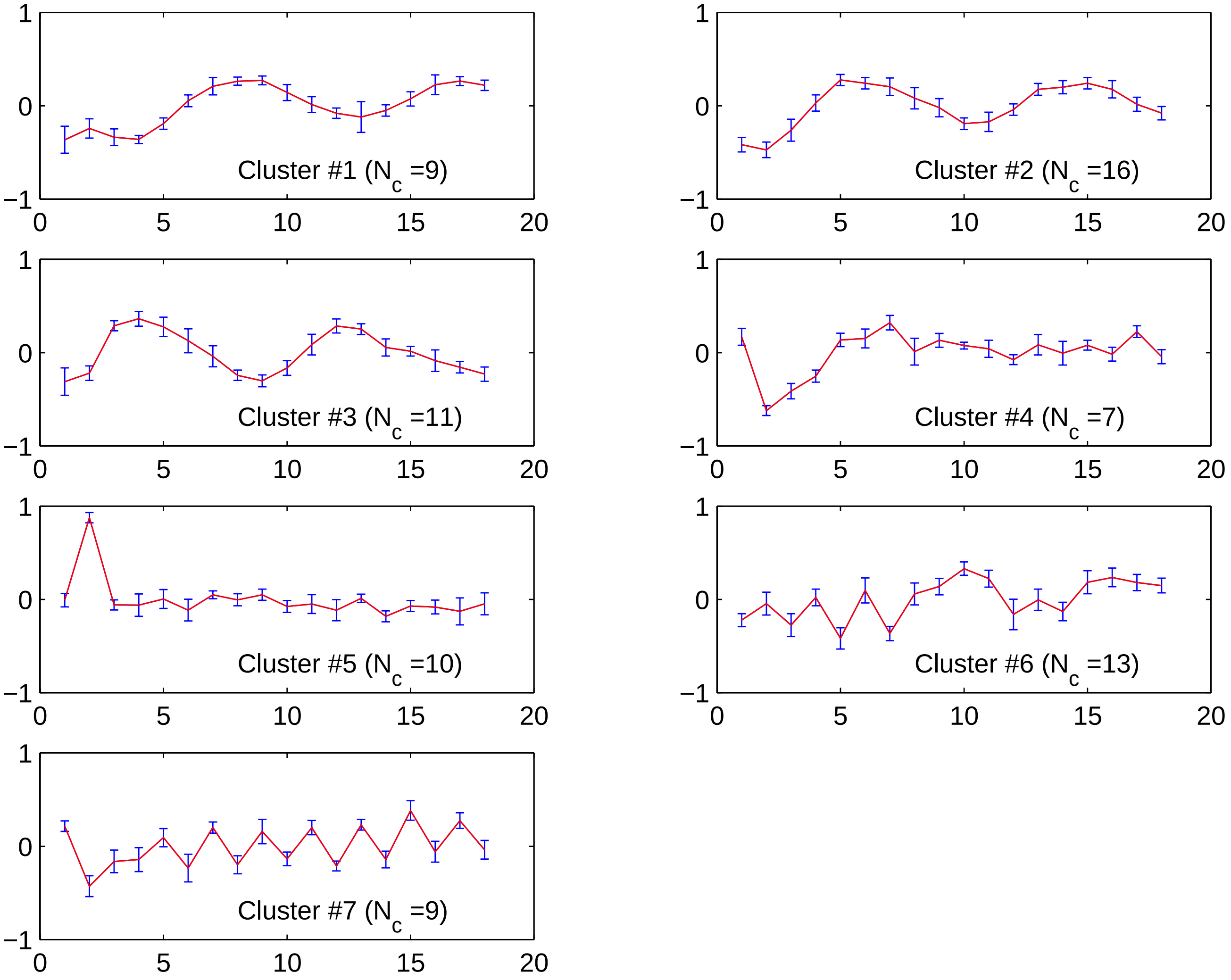


Figure A.6: Mean expression profile for the selected clusters. error bars represent the standard deviation of the expression of the genes in the cluster.

In summary, we find that the cluster validity index $\mathcal{Q}$ turns out to be rather efficient in identifying clusters of some biological meaning. The main advantage of this approach is the intuitive meaning of this index. Another benefit is that in fact the calculation of the index, and the thresholding, is not necessary: one can identify clusters of high $\mathcal{Q}$ scores just by looking at the dendrogram, and observind branches which remain stable through large range of temperatures.

Bibliography

[1] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data.* Prentice–Hall, Englewood Cliffs, 1988.

[2] I. Gath and A.B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Trans. Pat. Ana.*, 11:773–779, 1989.

[3] R.N. Dvae. Validating fuzzy partitions obtained through c-shells clustering. *Pattern recognition letters*, 17:613–623, 1996.

[4] P. Smyth. Clustering using monte carlo cross-validation. *KDD-96 Proceedings, Second International Conference on Knowledge Discovery and Data Mining*, pages 126–133, 1996.

[5] P.I. Good. *Resampling methods.* Springer-Verlag, New York, 1999.

[6] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall, 1993.

[7] K. Fukunaga. *Introduction to statistical Pattern Recognition.* Academic Press, San Diego, 1990.

[8] A.C.W May. Toward more meaningful hierarchical classification of protein three-dimensional structures. *Proteins*, 37:29–29, 1999.

[9] A.K. Jain and J.V. Moreau. Bootstrap technique in cluster analysis. *Pattern Recognition*, 20:547–569, 1986.

[10] E. Domany. Super-paramagnetic clustering of data – the definitive solution of an ill-posed problem. *Physica A*, 263:158, 1999.

[11] G. Getz, M. Vendruscolo, and E. Domany. Towards a fully automated protein structure classification: How to get cath classification from fssp z-scores. To Be Published.

[12] M. Blatt, S. Wiseman, and E. Domany. Super–paramagnetic clustering of data. *Physical Review Letters*, 76:3251–3255, 1996.

[13] M. Blatt, S. Wiseman, and E. Domany. Data clustering using a model granular magnet. *Neural Computation*, 9:1805–1842, 1997.

[14] S. Wiseman, M. Blatt, and E. Domany. Super–paramagnetic clustering of data. *Physical Review E*, 57:3767–3783, 1998.

[15] S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32:241, 1967.

[16] S. Wang and R.H. Swendsen. Cluster monte-carlo algorithms. *Physica A*, 167:565–579, 1990.

[17] N. Shental, E. Levine, and E Domany. Superparamgnetic clustering algorithm revisited. Unpublished.

[18] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley–Interscience, New York, 1973.

[19] K. Rose, E. Gurewitz, and G.C. Fox. Statistical mechanics and phase transitions in clustering. *Physical Review Letters*, 65:945–948, 1990.

[20] M. Schena et al. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270:467–470, 1995.

[21] M. Schena et al. Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. *Proc. Natl Acad. Sci. USA*, 93:10614–10619, 1996.

[22] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, 95:14863–14868, 1998.

[23] M. Chee et al. Accessing genetic information with high-density dna arrays. *Science*, 274:610–614, 1996.

[24] P.O. Brown and D. Botstein. Exploring the new world of the genome with dna microarrays. *Nature Genetics*, 21:33 – 37, 1999.

[25] E.S. Lander. Array of hope. *Nature Genetics*, 21:3–4, 1999.

[26] P. Tamayo et al. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, 96:2907, 1999.

[27] U. Alon et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA*, 96:6745–6750, 1999.

[28] D.E. Bassett, M.B. Eisen, and M.S. Boguski. Gene expression informatics -it's all in your mine. *Nature Genetics*, 21:51 – 55, 1999.

[29] G. Getz, E. Levine, and E. Domany. Dna data analysis. To Be Published.

[30] S. Chu, DeRisi, M. J., Eisen, et al. The transcriptional program of sporulation in budding yeast. *Science*, 282:699–705, 1998.

[31] P.T. Spellman et al. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.

[32] G. Getz, E. Levine, E. Domany, and M.Q. Zhang. Super-paramagnetic clustering of yeast gene expression profiles. physics/9911038.