A New Measure For Clustering Model Selection

by

Jesse McCrosky

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A new method for determining the number of k-means clusters in a given data set is presented. The algorithm is developed from a theoretical perspective and then its implementation is examined and compared to existing solutions.

Acknowledgements

I would like to thank my supervisors Shai Ben-David and Prabhakar Ragde for guiding me through the profound experience that my M.Math has been, as well as my readers, Alex Lopez-Ortiz and Ali Ghodsi, for helping me through the completion of this thesis. I would also like to thank the National Science and Engineering Research Council of Canada, the David R. Cheriton School of Computer Science, and the University of Waterloo Faculty of Matheatics for their generous financial support of my studies.

Finally I would like to thank all the wonderful people I have met here for making my time in Waterloo such a pleasurable and exciting experience.

Dedication

This thesis is dedicated to my parents, Carl and Judy McCrosky, in appreciation of the support they have offered me in all of my endeavours.

Contents

Chapter 1

Introduction

1.1 Overview

Clustering is an important data analysis tool that involves dividing a set of data into some number, k, of clusters that reflect natural groupings in the data set. We will consider the problem of k–means clustering and, specifically, the problem of determining a suitable value of k for a given data set. We will develop a framework for answering this question, the k–means Model Selection Problem, and consider desirable properties of a solution to this problem. We examine various solutions proposed in the literature before proposing and analyzing a novel approach. We give experimental results to show the effectiveness of the proposed method. Finally, we present further possible development of the method before offering conclusions.

1.2 Motivation

Clustering is a very important class of machine learning problems with wide-spread applications in bioinformatics and many other fields. In the most general sense, the clustering problem is, given a set of data points and a measure of distance between them, to partition the data points into some number of sets, such that the data points within each set (or 'cluster' or 'partition') are similar to each other and dissimilar from data points in other sets. This is an instance of the problem of unsupervised learning as studied in the field of artificial intelligence.

This definition, however, is broad, thus clustering is generally broken down into more specific problems. Among the most important is the k–means clustering problem. We will restrict our attention to this problem in the majority of this work; however, as discussed in Section 4.1, the results we obtain may be much more generally applicable.

1.2.1 How Many Clusters?

The k–means clustering problem has been thoroughly studied and many algorithms have been proposed; however, it is important to note that for the standard formulation of the problem the number, k, of clusters to partition the data into is given as input to the algorithm. In many cases, it is not easy to decide how many clusters should be used.

The naive solution would be to optimize the clustering cost function over the number of clusters. However, in this case, as we will see, the optimization puts each data point in its own cluster, which is clearly not an interesting solution. Some notion of structure in the data set must be analyzed in order to algorithmically determine the 'correct' value of k.

In the majority of real-world clustering applications, the appropriate value of k is not known in advance and so must be guessed or found using some heuristic. Clearly, this is an important problem in an applied sense, but it is also important theoretically: understanding how to determine the number of clusters in a data set is an important step towards thoroughly understanding the structure that clustering is intended to extract.

1.3 Problem Definition

We are interested in finding the number of clusters in a given data set. There is more than one possible way to formulate this problem, and we will consider two possibilities. First, however, we must formalize the clustering problem itself.

1.3.1 The k–means Problem

The k–means clustering problem is among the most important and thoroughly studied forms of clustering. The term 'k–means clustering' is often also used to describe certain algorithms for solving the k–means problem, but we are not concerned here with how the problem is solved.

We consider the k–means clustering problem on d-dimensional Euclidean space. The input of the problem is a data set, $X = \{x_1, x_2, \ldots, x_n\}$, $x_i \in \mathbb{R}^d$, and an positive integer, $k \leq n$. The problem is to find a partitioning which will minimize the k–means cost function given below.

A k–partitioning, $\mathcal{C} = \{c_1, c_2, \ldots, c_k\},\ c_i \in 2^X$, is defined as a set of partitions, each of which is a set of data points. The definition of a k–partitioning requires that every element of the data set is contained in one of the partitions and no partition contains an element not in the data set:

$$\bigcup_{c \in \mathcal{C}} c = X$$

and that, assuming all data points are unique, the partitions are pairwise disjoint:

$$\forall_{c_\alpha, c_\beta \in \mathcal{C}} \, c_\alpha \neq c_\beta \rightarrow c_\alpha \cap c_\beta = \emptyset.$$

This definition satisfies are intuitive sense of what it means to divide a data set into k separate groups. However, there are many such divisions possible, so we use the k–means cost function to choose the 'best' such partitioning.

The k–means cost function that we wish for our partitioning to minimize is:

$$R(\mathcal{C}, k) = \sum_{c \in \mathcal{C}} \sum_{x \in c} ||x - \overline{c}||^2$$

where $\overline{c} = \frac{1}{|c|} \sum_{x \in c} x$ is the center of mass of the partition c, and $|| \cdot ||$ is the Euclidean distance.

We can now define the optimal k–partitioning of a data set as the k–partitioning having the lowest k–means cost. This optimal solution will not always be unique.

1.3.1.1 Properties of the k–means Cost Function

The following properties of the k–means cost function will be important later in our discussion:

- Scale Proportionality: If a data set is uniformly scaled by a constant, c, then the cost of any given partitioning of that data set will be scaled by c^2. This is because the distances will all increase by a factor of c and the cost function is proportionate to the square of the distances. That is, for all X and k:

  $$R(\mathcal{C}_c, k) = c^2 C(\mathcal{C}_1, k)$$

  where $\mathcal{C}_1$ is any partitioning of X and $\mathcal{C}_c$ is the equivalent partitioning of $c \times X$.

- Monotonic Decreasing with k: For any fixed data set, the k–means cost of an optimal k–partitioning will be non-increasing as k increases. This is intuitively reasonable, as increasing the number of partitions will allow for smaller, and thus less costly, clusters.

1.3.2 The Meta-problem

Now we consider the problem of determining the correct number of clusters (or value of k) for a given data set. This problem is ignored in the standard formulation of k–means, but must be solved before any k–means algorithm can be usefully applied.

We are interested in the correct number of clusters in the context of the k–means objective. The clusters that are found in k–means have specific properties (for example, generally, they are convex) that may differ from other clustering objectives. The number of clusters in a data set may differ under different clustering objectives. From here, we will simply use the term 'cluster' to refer to a k–means style cluster.

1.3.2.1 The k–means Number of Clusters Problem

The k–means Number of Clusters Problem is, given a data set, to determine how many clusters exist in the data set. The problem can be formalized as attempting to find a function:

$$k(X) :, \mathcal{X} \to \mathbb{Z}^+$$

where $\mathcal{X}$ is the set of all possible data sets on the domain under consideration, i.e. $\mathcal{X}$ is the power set of $\mathbb{R}^d$ for whichever value of d is being considered. We will consider the properties we wish this function to have in Section 1.4.

The function will map any data set to the number of clusters that exist in that data set. This definition, however, hides the real difficulty: in many cases the correct number of clusters is ambiguous. There may be multiple numbers that can be considered correct, perhaps depending on context. We will discuss this problem in further detail below. First, however, we consider an alternative formulation of the problem that allows for slightly more flexibility.

1.3.2.2 The k–means Model Selection Problem

Our alternative formulation is, given a data set and a k value, to determine the degree (the Model Fit Value) to which the given k value fits the given data set. This problem allows for a much more flexible view than the Number of Clusters Problem and, as we will see in Section 4.1, can be generalized in interesting ways.

In the context of this problem, each possible value of k represents a clustering model. The problem is then to determine how well the model fits the data. If we wish to find the correct value of k for a data set, we must choose a set of prospective models that includes the correct model and evaluate the Model Selection Measure on each of them. The model that evaluates to the greatest Model Fit Value is the correct model. In some cases there may be more than one correct model. The problem can be formalized as the search for a function:

$$\mathcal{M} : \mathcal{X} \times \mathbb{Z}^+ \to \mathbb{R}$$

such that, for a fixed data set, the value of the function will be greater for values of k that better fit that data set. We will discuss the properties we desire this function to have in more detail in Section 1.4.

1.3.2.3 The General k Problem

We have presented two different formalizations of the problem of determining the number of clusters in a data set; however, it is sometimes useful to consider the problem in a more general sense. We shall refer to this problem in general as 'the k problem'.

1.3.2.4 Differences Between the Number of Clusters Problem and the Model Selection Problem

Although they solve the same general problem, the Number of Clusters Problem and the Model Selection Problem are different in important ways. An instance X of the Number of Clusters Problem can be reduced to n instances $\{\langle X, 1\rangle, \langle X, 2\rangle, \ldots, \langle X, n\rangle\}$ of the Model Selection Problem by returning the value of k which results in the maximal Model Fit Value among the Model Selection Problem instances.

No reduction is possible in the opposite direction, however, as the Model Selection Problem is more flexible than the Number of Clusters Problem. As we will consider in Section 4.1, the Model Selection Problem might also be extended in some interesting ways to provide capabilities further beyond the Number of Clusters Problem.

We will be primarily interested in the Model Selection Problem in the remainder of this thesis.

1.3.2.5 The Fundamental Ambiguity of the Meta-problem

It is important to emphasize that, asked to find the 'correct' value of k for a data set, the problem is to *define* the correct value. There is no intrinsically correct value of k for any data set, thus any reference to a 'true' or 'correct' value is inherently subjective. We will refer to 'correct' solutions and, as we see below in Section 1.4, there are strong arguments for what value of k should be correct for certain data sets; however, the reader is asked to remember that the use of the terms 'true' or 'correct' in reference to a value of k is a stretch of the terminology.

1.3.2.6 Terminology

When discussing the k problem, we must consider how a data set might be clustered for various different numbers of clusters and compare these clusterings. Because of this, there are some concepts that are important to be able to describe succinctly, so we will fix some terminology here. For any given data set, we may refer to a 'correct' number of clusters, which will be taken as an absolute and supersedes the judgment of any other method for determining k. Thus, if we have fixed a correct solution for a given data set and a k–determination method under consideration gives a different answer on that data set, we will refer to that solution as being 'incorrect' on that data set. There may, however, be more than one correct value of k as in Figure 1.1, in which both 2 and 4 could be correct.

We will refer to these correct values of k as follows: for a data set X for which we can define one or more correct numbers of clusters, we will denote this set of correct values of k as $k^*(X)$.

Relative to a particular value of k we consider to be correct, we will refer to 'true' clusters, those obtained under an optimal k–means clustering of a data set with that 'correct' value of k (the 'true' clustering). Any other cluster is a 'false' cluster or part of a 'false' clustering.

Now we can define the intra–cluster and inter–cluster distances:

- The intra–cluster distances are those distances between points that are in the same true cluster, i.e. they are clustered together in the optimal k–clustering for the correct value of k considered.
- The inter–cluster distances are those distances between points that are in different true clusters, i.e. they are not clustered together in the optimal k–clustering for the correct value of k considered.

By the definition of the k–means clustering objective, intra–cluster costs will tend to be be small. In typical data sets in which there is separation between clusters, inter–cluster costs will tend to be large.

1.4 Desired Properties of a Solution

In this section, we give some general properties we desire of a Model Selection Measure as well as considering special classes of data sets that can be used for evaluating a measure; these will be data sets for which we can make strong arguments regarding the correct number of clusters.

1.4.1 Scale Invariance

First and most simply, we ask that, if a particular data set, X, has k clusters, then any uniform scaling of that data set, $\alpha \times X$, should also have k clusters. This is a consequence of one of the basic properties that might be desired of a clustering function as proposed by Kleinberg in [12]: that the clustering of a data set should be invariant under uniform scaling of the data.

1.4.2 Multiple Solutions

Secondly, we ask that a measure be capable of considering multiple good values of k for a particular data set. For example, in Figure 1.1, we see a data set for which either 2 or 4 is a reasonable number of clusters. We formalize this property as follows:

Defining a local optimum for data set X as a value of k such that

$$\mathcal{M}(X,\, k) > \mathcal{M}(X,\, k-1) \text{ and } \mathcal{M}(X,\, k) > \mathcal{M}(X,\, k+1),$$

we ask that a measure allow for multiple local optima for some data sets such as that in Figure 1.1.

1.4.3 Large Separation and Human Clustering

We now consider criteria which allow us to argue for the correct value of the number of clusters in certain data sets. The first such case we consider is powerful but fundamentally subjective: the opinion of a human observer. Clustering is one of the fundamental operations of the human brain, and we are very good at it [6]. Thus, in many cases, it is simple for an observer to assess the number of clusters in a data set merely from observing a graphical representation of that data set, especially when there is large separation between the clusters.

This method is, however, limited. Typically, human clustering is possible only in the case of a one or two dimensional data space. Another weakness is that we are interested here in k–means clustering; however, the way the human brain clusters is often different. As such, there may be data sets for which the number of clusters according to the human clustering algorithm is not the correct number of clusters for the k–means clustering algorithm. Due to these weaknesses, it is difficult to apply the human clustering criteria and so we will consider some more formal criteria instead.

1.4.4 Perfect Clusterings

The human clustering approach generally relies on there being sufficient separation between clusters as to make the divisions unambiguous. This unambiguous division can be formalized: In [7], the notion of a 'perfect clustering' is introduced. A clustering is perfect if the maximum distance between any two data points in the same cluster is less than the minimum distance between any two data points in different clusters. We can use this notion as a criteria for membership in $k^*(X)$:

- If, for a given data set, X, and a given value of k, there exists a perfect k-clustering of X, then we say that $k \in k^*(X)$.

It should be made clear that the existence of a perfect clustering is sufficient to justify the membership of a particular value of k in $k^*(X)$, but it does not suffice to rule out any other values of k. For example, the data set given in Figure 1.1 is perfectly clusterable for $k = 4$ and for $k = 2$. It should also be noted that the absence of a perfect clustering does not necessarily invalidate a prospective number of clusters: a perfect k–clustering of X is a sufficient, but not necessary, criterion for membership in $k^*(X)$.

1.4.5 The k–defining Distributions

We can also consider distributions over our data domain which can be said to have a inherent justifiable number of clusters. It is generally not possible to claim that all data sets sampled from a given distribution will have a particular number of clusters; however, we often can argue that, with high probability, any sufficiently large sample will have the number of clusters associated with the distribution. We will handle this with a failure

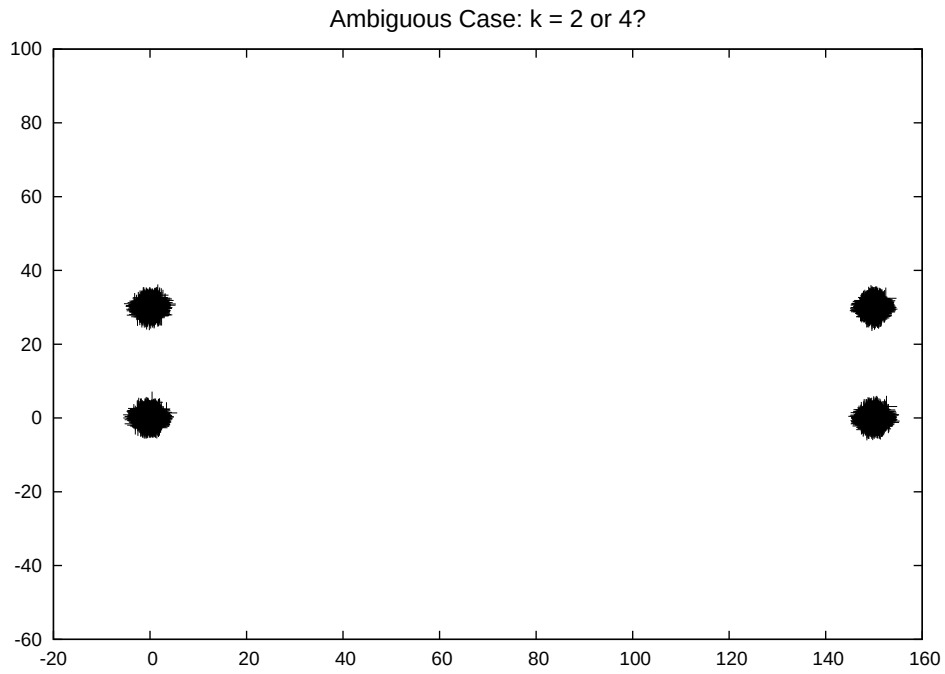

Figure 1.1: Data Set with ambiguous k

probability function, $\epsilon(n_0)$ which, for a minimum sample size n_0, specifies the maximum probability with which a measure should choose a value of k other than that associated with the distribution.

More formally, if G is a k–defining distribution with number of clusters k and failure probability function $\epsilon(n_o)$, we ask that a measure, $\mathcal{M}$, have probability less than or equal to $\epsilon(n_0)$ of choosing a number of clusters other than k for any sample of at least n_0 data points That is, for all $n > n_0$:

$$\Pr_{X \sim Samp_n(G)} \left(\exists_{k'} \; \mathcal{M}(X, k) < \mathcal{M}(X, k') \right) \leq \epsilon(n_0)$$

where $Samp_n(G)$indicates a sample of n data points from G.

Having described the way a measure should behave on a k–defining distribution, we now consider what distributions can be considered k–defining.

1.4.5.1 Mixture Distributions

A Mixture Distribution is simply a weighted sum of two or more component distributions, such that the total weight of all components sums to 1. The PDF of a mixture distribution is:

$$f_{mix}(x) = \sum_{i=1}^{j} w_i f_i(x)$$

where j is the number of components in the mixture, $f_i(x)$ is the PDF of the i^{th}component of the mixture, and w_i is the weight of the i^{th} component where $\sum_{i=1}^{j} w_i = 1$. We will refer to a Mixture Distribution with $w_i = \frac{1}{j}$ as a Uniform Mixture Distribution.

We will refer to a Mixture Distribution that is k–defining for the number of components in the mixture ($k = j$) as a k–defining Mixture Distribution.

1.4.5.2 Examples

As an example, consider a uniform mixture distribution of two uniform distributions on $\mathbb{R}$. One of the uniform distributions is over $[-5, -4]$ and the other is over $[4, 5]$. The PDF of this mixture distribution will be:

$$f(x) = \begin{cases} \frac{1}{2} & \text{where } x \in [-5, -4] \cup [4, 5] \\ 0 & \text{otherwise.} \end{cases}$$

In this case we can clearly see (using the human clustering criteria) that there is a very high probability that for any large data set sampled from this distribution, $k = 2$ will be justifiable as the number of components in the mixture. In fact, a Perfect Clustering will exist in almost all cases, making the choice even easier to justify. If the data set is small, however, there will be a non-negligible probability that all of the sample data points will be from only one of the two components. It is likely that $k = 2$ will not be justifiable for these data sets. Thus, a sufficient sample size is necessary and, even for an arbitrarily large sample size, there is a non-zero probability that our data set will not have $k = 2$ as desired. It is clear that there are limitations to k–defining distributions; these limitations are characterized by the $\epsilon(n_0)$ function. If we assume that the existence of a single data point in each interval is sufficient to establish the existence of two clusters (a questionable assumption, but such assumptions are typically necessary to precisely define $\epsilon(n_0)$), then $\epsilon(n_0) = 2^{(1-n_0)}$ in this case.

As another example, consider a Uniform Mixture Distribution of three Uniform Distributions on $\mathbb{R}$. One of the Uniform Distributions is over $[-5, -4]$, one is over $[4, 5]$ and the other is over $[4.1, 5.1]$. The PDF of this distribution will be:

$$f(x) = \begin{cases} \frac{1}{3} & \text{where } x \in [-5, -4] \cup (4, 4.1) \cup (5, 5.1) \\ \frac{2}{3} & \text{where } x \in [4.1, 5] \\ 0 & \text{otherwise.} \end{cases}$$

In this case, despite the fact that we have three components in the mixture distribution, any data set sampled from this distribution is likely to have $k = 2$ and not $k = 3$. This is because of the intersecting regions of two the components. Separation between the regions of each of the components is, as we will see, a basic requirement on a mixture distribution for it to be k–defining. In this case, although it is possible that a particular data set sampled from this distribution might have $k = 3$, as the sample size increases, the distribution will be increasingly likely to have only two clusters and thus, the probability

of a correct measure choosing a value other than three will converge to 1. Thus, this distribution is k–defining only in the pathological case with $\epsilon(n_0) = 1$.

1.4.5.3 Mixture Distributions of Convex Multivariate Uniform Distributions

One extremely simple class of k–defining distributions is the Mixture Distribution of Convex Multivariate Uniform Distributions (MDCMUD). A Convex Multivariate Uniform Distribution assigns equal probability distribution over some convex subset of the data domain; we will refer to this subset as the region of the distribution. A MDCMUD is simply a Mixture Distribution of these.

We will define the minimum component separation of a MDCMUD as the minimal distance between any two points in the regions of different components of the distribution; in the case of overlapping regions, this distance may be negative.

Assuming the minimal component separation of the distribution is greater than 0, a MDCMUD is a k–defining Mixture Distribution. It is simple to show that, for these distributions, the probability that a sample has a perfect k–clustering converges to 1 as the size of the sample, n, approaches infinity; this is because, if we map components to clusters, by the definition of minimal component separation, the minimal distance between points in different clusters in greater than 0, but the maximum distance between points in the same cluster will converge to 0 as n approaches infinity.

Thus we have a large class of k–defining Mixture Distributions. With larger minimal component separations, the $\epsilon(n_0)$ function will be small even for relatively low values of n_0, while if the components are very close, this function will tend to be high.

1.4.5.4 Mixture Distributions of Multivariate Gaussians

Another important class of Mixture Distributions is the Mixture Distribution of Multivariate Gaussians (MDMG). We consider this class as it is commonly used in the literature for testing solutions to the k problem and because there are some theoretical results on these distributions that gives us some insight into our error probability function, $\epsilon(n_0)$.

A Multivariate Gaussian or Multivariate Normal random variable is defined by the PDF:

$$\frac{1}{(2\pi)^{\frac{k}{2}}\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$$

where μ is a vector indicating the mean of the distribution, and Σ is the covariance matrix which determines the shape and dispersion of the distribution.

Using this definition, the probability distribution function of a MDMG is:

$$\sum_{i=1}^{k} \frac{w_i}{(2\pi)^{\frac{k}{2}}\sqrt{|\Sigma_i|}} \exp\left(-\frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i)\right)$$

where $w_i$is the relative weight of each component, μ_i is the mean of each component, and Σ_i is the covariance matrix of each component.

There are good reasons why these models are frequently used in the literature for testing of clustering and k–determination methods. The class of models is rich enough to approximate many realistic data sets but simple enough to be easily parameterized. Further, the maximum likelihood over these models (usually with all parameters other than the means fixed) is another popular clustering objective function, quite similar to the k–means objective function. In fact, for a uniform mixture of spherical Gaussians, the maximum likelihood objective converges to the k–means objective as the variance of the Gaussians approaches 0. Later, we will use these models in our experiments; first, however, we wish to consider in what cases these distributions are k–defining and what factors determine the error probability function $\epsilon(n_0)$.

1.4.5.5 Theoretical Limitations

It can be seen that with insufficient samples, a measure can not be expected to find k clusters in a data set sampled from a k–determining MDMG. As a trivial example, consider a sample of two data points from a three–component MDMG.

Likewise, if the means of two of the components are too close, the distribution might have a very high error probability function. For two Multivariate Gaussians with means $\mu_1$and μ_2 and covariance σI for both, we can define the separation between them as:

$$\frac{\|\mu_1 - \mu_2\|}{\sigma}.$$

In the case of non-spherical Gaussians or Gaussians with different covariances, the definition is more complex. Given this definition, we can now discuss the minimum pairwise separation (minimal among all pairs) of the components of an MDMG. It may also be interesting to consider average pairwise separation or other similar measures as well.

In general, this minimum pairwise separation must be sufficiently large in order for an MDMG to be k–defining. If two clusters are too close, it becomes reasonable for their samples to be clustered together, reducing the correct number of clusters to less than the number of components in the MDMG.

In order to consider these limitations, we will consider some work on a similar problem and extrapolate the results. In [23], the problem of learning the centers of a MDMG is considered, with a known number of components. It is shown that there is a theoretical limit in the required sample size and required component separation for a MDMG to be learnable with high probability. These limits are more restrictive with higher numbers of components and higher dimensionality. Although these results apply to learning the means only, the results can be extrapolated to the case of learning the number of components: in the learning problem of [23], the means of the Gaussians are being learned; in our problem, it is simply a different parameter, the number of components in the mixture, that is being learned.

In general, we can say that a MDMG is only k-defining if there is sufficient separation between its components. The separation required and number of samples required to keep the error probability low will increase as the number of components and the dimensionality of the space increases. The actual requirements will be explored experimentally in Chapter 3.

As an aside, it is also useful to note that in [23], it is found that there exists a regime of MDMG models for which accurate learning is theoretically possible, but not tractable. Thus it is likely that there is a regime of models which are theoretically k–defining, but no efficient algorithm will find the correct number of clusters consistently. This will be an important limit to consider in the implementation of our Model Selection Measure.

1.4.5.6 Conclusions of k–defining distributions

We have seen that data sampled from a k–defining distribution should be found to have k clusters with high probability, but it is generally not possible to specify exactly what this probability is. As such, this property is valuable only in comparing different methods of k–determination: The method with the lower error probability function on a particular distribution can be said to handle that distribution better.

1.4.6 Conclusions of the Desired Properties

We have given a number of properties here that we wish a Model Selection Measure to possess. To summarize, a measure should:

- be scale invariant,
- allow for multiple solutions,
- accept k as a solution whenever there is a perfect k–clustering of the data set,
- accept k as a solution with probability at least $1 - \epsilon(n_0)$ for a sample of at least n_0 data points from a k–determining distribution with small as possible error probability function $\epsilon(n_0)$.

We will use these properties to evaluate previously proposed solutions as well as our new solution proposed below.

1.5 Previously Proposed Solutions

In general, the previously proposed methods for determining the k problem fit into one of two categories: those that analyze the 'cost versus k' tradeoff on the data and those that analyze the stability of the optimal k–means clustering of the data under various perturbations. We describe a representative implementation from each class below.

1.5.1 Cost Versus k

It is generally well accepted, as in [11], that analyzing the rate at which the k–means cost of the optimal k-clustering decreases while k increases is a fairly effective heuristic for finding an appropriate value of k. Generally, the cost will decrease dramatically as k increases up to the correct number of clusters, but will decrease only slowly after that point. In the case of more than one correct number of clusters, there will be multiple points at which the cost's rate of decrease lessens, although in these cases, the transitions can be more difficult to detect.

In order to understand the reasoning behind this heuristic, recall the terminology presented in Section 1.3.2.6 and consider that a data set is made up of k^* 'true' clusters. The distances between any two data points within a true cluster (the intra–cluster distances) should be relatively small and the distances between any two data point in different true clusters (the inter–cluster distances) should be relatively large.

The intuition behind the heuristic is that, up to the correct value, adding another cluster should substantially improve the cost, as the increase in k will allow two true clusters that were previously incorrectly combined to be separated, removing some large inter–cluster distances from the cost of the clustering. However, once the correct k value is reached, there will be little benefit to adding additional clusters, as this will only serve to split one or more true clusters, removing only the relatively small intra–cluster distances from the cost.

Thus, the heuristic states that the correct value of k is that at which the decrease in cost sharply flattens. This heuristic can be quite effective, but is subjective and difficult to formalize. Despite this, many of the successful heuristics used are based on this observation. Perhaps the most effective such heuristic is the Gap Statistic.

1.5.1.1 The Gap Statistic

The Gap Statistic, proposed in [11], is among the best-known and most effective heuristics for finding the correct number of clusters in a data set. It was designed as a direct formalization of the cost versus k heuristic, described in Section 1.5.1 above, but has some interesting differences. Briefly, the algorithm compares the cost of the optimal k–means cost on the data set for each value of k considered to the optimal k–means cost of a 'reference', uniform distribution. The k for which the cost on the data has the greatest advantage over the cost on the reference distribution is considered the correct value of k. This technique is especially interesting as it includes heuristics to consider the hypothesis that the data is unclusterable ($k = 1$), a feature missing from many other methods. Simply stated, the heuristic will select $k = 1$ if no greater value of k has a sufficient difference in cost between the data set and the reference distribution.

There are two variation of the method presented. In the "uniform" version, the reference distribution is distributed uniformly in the minimal axis-aligned box containing the data; in the "principal component" variation the box is aligned with the principal component of the data.

The Gap Statistic is generally fairly effective; however, it will not consistently perform as expected even in relatively easy problems and, especially, will often inappropriately conclude an absence of structure ($k = 1$). For example as shown in [14], the Gap Statistic incorrectly concluded $k = 1$ for a mixture of 5 well-separated Gaussians.

Generally, the Gap Statistic meets all of our criteria of Section 1.4 except that, as proposed in [11], it is not capable of handling multiple solutions and will fail on some k–defining distributions with a worse than necessary error probability function as we will discuss in Section 3.4.1.

1.5.2 Stability Methods

Stability is another common approach for solving the k problem or, more generally, evaluating whether or not a clustering model is correct on a given data set. There is a large body of work on these methods, including [3] and [14]. The general principle of the stability methods is that a model fits a data set well if a specified algorithm will find approximately the same clustering of that data set even if the data is perturbed in some way, such as by taking various partial samples of the data set or adding noise to the data.

These methods are often very effective; however, as explained in [2], their effectiveness may be misleading. The paper shows that stability under sampling is dependent only on the existence of a unique optimal solution to the clustering cost function. It can be argued that a correct clustering model should induce a unique optimal solution; however, this should not be taken as an absolute requirement. Consider a data set with two well-separated clusters and a single data point centered between the clusters. It is possible to form arguments for the correct number of clusters bring both 2 and 3. However, $k = 2$ does not have a unique optimal solution, as the center point can be assigned to either cluster without affecting the cost. See Section 3.3.1.8 for experimental analysis of this situation. Unless we wish to disallow $k = 2$ as a solution for this case, we can not take uniqueness of optimal solution as an absolute requirement for a correct model.

It is also important to note that incorrect models may also induce unique optimum solutions, as shown in [2]. Thus the existence of a unique optimal solution can, at most, be taken as a necessary, but insufficient criterion for the correct clustering model.

Stability under perturbation of the data set has also been shown to be an effective measure [3]; it is likely, however, that it is also determined by the existence of a unique optimal solution.

In general, stability methods can meet all of the criteria discussed in Section 1.4; it is only cases discussed above in which correct models have multiple optimal solutions or incorrect models have a unique optimal solution in which the methods tend to fail.

1.5.2.1 The Swiss Stability Method

The implementation of the Stability Method given in [14] is an excellent example of such methods. It determines the degree to which a given data set X can be clustered into

k clusters by determining how similarly two subsets of X are clustered. In detail, X is split into two non–intersecting subsets, X_a and X_b. The clustering algorithm being used is applied to X_a finding a labeling for each element of that set. This labeling is used to train a classifier. X_b is then clustered using both the clustering algorithm and the classifier trained on X_a and the difference between these two labelings (minimized under relabeling) is used as a measure of stability for the clustering method, X and k: if the two cluserings are very similar, the model fit is good.

Chapter 2

Entropy-based k–means Model Selection

2.1 Overview

In this chapter we propose a new k–means Model Selection Measure based on the Renyi entropy [21] of a random variable we construct to represent the possible k-clusterings of the data set. This method is completely different from existing methods in that it does not rely on either the cost versus k heuristic nor any stability properties. We will first consider the goal we wish to meet with our measure. We will then proceed to consider the available information that may allow us to reach this goal, and then consider how the information might be interpreted and applied. Once we have found a possible method, we will formalize this method and then present the obtained measure before evaluating it and comparing it to other existing methods.

2.2 The Goal

We wish to find a k–means Model Selection Measure, which we will define as a function:

$$\mathcal{M} : \mathcal{X} \times \mathbb{Z}^+ \to \mathbb{R}.$$

This function will be defined in detail in the remainder of this section. We will evaluate the measure by considering the desired properties given in Section 1.4.

2.3 Available Information

The only information we are explicitly given is the value of k to be evaluated and the data set itself. Combining these two elements and considering the k–means cost function as well, we find that we also have access to the k–means cost of each possible k–partitioning of the data, as well as the k–partitionings themselves.

2.4 Interpretation of the Information

This information can, of course, be interpreted in many different ways. It is intuitively satisfying, however, to imagine a distribution of the cost over a space of possible partitionings with similar partitionings close to each other and dissimilar partitionings distant from each other.

This cost distribution over the partitionings is interesting, but in order to simplify our explanation, we will define a new, similar, distribution. Instead of considering the k–means cost of each partitioning, we will consider a normalized value, related to the cost. Given a set of partitionings, $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$, we can find the k–means cost, $R(\mathcal{C}, k)$, for each. We will then define the value of each partitioning to be the normalized reciprocal of the cost, $V(\mathcal{C}, k) = \frac{R(\mathcal{C},k)^{-1}}{\sum_{\mathcal{C}_i \in \mathcal{P}} R(\mathcal{C}_i,k)^{-1}}$. We choose to take the reciprocal in order to ensure that 'better' partitionings have higher values, rather than lower. This distribution will be called the Partitioning Value Distribution.

We can now consider the characteristics of the distribution. We will see peaks at the locations of the 'good' clusterings. These peaks will typically not be single-element spikes as, usually, given a good clustering, there will be many similar clusterings that are almost as good, perhaps varying only in the assignment of a single data point.

This distribution can be said to characterize how the particular clustering model applies to the given data set. Careful examination of it can decide the degree to which the model fits the data.

2.5 Application of the Information

In order to determine how this distribution can be used to determine the model fit, we will consider how the Partitioning Value Distribution might look in three cases. It is assumed in all cases that the data set has a clearly defined number of clusters. It is important to recall the terminology explained in Section 1.3.2.6 as it is used here extensively. It is also important to emphasize that the 'correct' number of clusters is loosely defined. There are data sets for which there are multiple possible values of k that can be justified. Thus, for a data set that can reasonably be clustered into either two or four clusters, the case of $k = 3$ is, in a sense, an underestimation and an overestimation simultaneously, thus both cases will apply to some extent.

It should also be mentioned that, in these cases in which the number of clusters is clearly defined, the inter–cluster distances will tend to be larger than the inter–cluster distances. This is necessary in order to create the separation that distinguishes the clusters.

2.5.1 Partitioning Value Distribution in Case with k Underestimated

If k is underestimated, all partitionings will be forced to include elements from multiple true partitions in a single false partition. This will result in even the best partitionings being little better than the average partitioning as they will include inter–cluster costs. Under the general assumption that intra–cluster costs are substantially smaller than inter–cluster costs, any partitioning that includes some inter–cluster costs will be substantially worse than the true partitioning, which includes only intra–cluster costs.

Thus, the distribution will appear relatively flat, with many slightly higher plateaus representing cases where the number of inter–cluster costs paid is relatively minimal.

2.5.2 Partitioning Value Distribution in Case with k Correct

Often, if the value of k is correct there will be a unique optimal solution. In this case, there will be a single large peak in the distribution. Even if there are multiple optimal solutions, there will generally be few of them and they will probably be very similar. The peak (or peaks) will be quite sharp, as any deviation from the optimal solution will begin to include the expensive inter–cluster costs.

Thus, in this case, the distribution will be sharply peaked.

2.5.3 Partitioning Value Distribution in Case with k Overestimated

If k is overestimated, there will be a large number of solutions with near optimal cost. The excessive partitions may result in one or more true partitions being split; however, this splitting will only affect the (small) intra–cluster costs and will thus not change the overall cost substantially. Each partition can be split in many different ways, thus each possible set of split partitions will generate a plateau of partitions with cost approximately equal to the optimal.

Thus, the distribution will include a large number of plateaus, each of approximately the same height.

2.5.4 Summary

We can see that there is generally a detectable difference in our distribution in each of these three cases. As described, these differences are qualitative and somewhat subjective, so it remains to find a method to detect each of these three cases algorithmically.

2.6 Formalization

We wish to define a function that will exploit these characteristics in order to perform the desired model selection. The characteristics are only generally defined, and so there is still some intuition needed in finding an appropriate function.

2.6.1 Throwing Away Information

We can immediately see that, in the case that k is correct, there will be a peak with a much better cost than most other parts of the distribution. This recognition leads us to our first possible class of functions we consider, but first it is interesting to note that this observation depends only on the values of the partitions, not the partitions themselves. We have found that an effective measure can be formulated using only the set of values of the partitionings, ignoring the partitionings themselves; we can consider only the Partitioning Values, $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$, not the Partitioning Value Distribution. Although it is possible that a better measure could be found by incorporating partitioning information (see Section 4.3), we believe that our value–only measure is very effective and is much more efficient because distances between partitionings need not be calculated.

Given this restriction, we can reformulate or measure to take a set of partitioning values as its single parameter:

$$\mathcal{M} : \mathcal{V} \to \mathbb{R},\ \mathcal{V} = \{v_1, v_2, \ldots, v_m\}.$$

Note that $\mathcal{V}$ will (implicitly) be a function of X and k.

2.6.2 Simple Statistics: Minimum and Maximum Values

Our recognition that a high peak is an important factor in determining the correct model suggests that the maximum value obtained is an important statistic (remembering that costs are normalized). We could propose that the difference between the maximum value obtained and the minimum or perhaps average value might be a useful measure. The ratio between the maximum and minimum or average might also be considered. Empirically, the functions were all found to be somewhat effective on the relatively easy problems considered, but to varying degree. From best to worst, they were:

1. $\mathcal{M}(\mathcal{V}) = \frac{\max(\mathcal{V})}{\text{avg}(\mathcal{V})}$
2. $\mathcal{M}(\mathcal{V}) = \max(\mathcal{V}) - \text{avg}(\mathcal{V})$
3. $\mathcal{M}(\mathcal{V}) = \max(\mathcal{V}) - \min(\mathcal{V})$
4. $\mathcal{M}(\mathcal{V}) = \frac{\max(\mathcal{V})}{\min(\mathcal{V})}$

Comparing the maximum to the average value is much more effective than comparing it to the minimum value. Still, it seems it should be possible to do even better. In considering only the maximum, minimum, and average cases, we lose a lot of the information that we have available, such as whether there are many partitionings with near the maximum value or only one.

2.6.3 Better Statistics: Entropy

2.6.3.1 Intuition

Intuitively, it is very natural to consider measures of entropy as prospective functions for our Model Selection Measure. Entropy is normally a measure on probability distributions only, but we will ignore this fact for the moment. As we discussed in Section 2.5, we are trying to find how sharply peaked (although not necessarily just a single peak) our distribution is; entropy can measure this. Entropy, of course, would be inversely related to the quantity we are searching for: a more sharply peaked, and thus more clusterable, distribution will have lower entropy.

2.6.3.2 How to Apply Entropy

Entropy is a measure on probability distributions, not Partitioning Value Distributions. However, due to our normalization, our Partitioning Value Distribution sums to 1, as must a probability distribution. To be clear, it can't be claimed that the Partitioning Value Distribution actually is a probability distribution; however, the similarity is sufficient for us to proceed and see what happens when we attempt to measure the entropy of our Partitioning Value Distribution.

2.6.3.3 Definitions of Entropy

The best known entropy measure is Shannon Entropy; however, as mentioned in [17], there are other possible measures of entropy we may consider. We evaluated Shannon Entropy, Renyi Entropy, and Burg entropy.

Given a random variable X with outcome probabilities p_1, p_2, ..., p_n, we can define each of our forms of entropy. Classical Shannon entropy, as introduced in [22], is defined as :

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i.$$

Shannon gives three basic properties that a measure of entropy should satisfy:

1. The function should be continuous on the probabilities.

2. If the probabilities are equal, the function should be monotone increasing with the number of possible outcomes.

3. If the outcomes can be broken down into a multi-stage decision, the entropy of the entire process should be a weighted (by probability) sum of the entropies of the subprocesses.

Shannon goes on to prove that the only functions that can satisfy these requirements will be constant scalings of his definition. However, there are still other useful entropy functions.

Renyi entropy, introduced in [21], is defined as:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n} p_i^\alpha \right).$$

As can be seen, Renyi entropy is actually a class of entropy functions, as parameterized by α. Renyi shows that in the limit as $\alpha \to 1$, the Renyi entropy approaches the Shannon entropy, thus, Renyi entropy can be seen as a generalization of Shannon entropy.

Burg entropy is defined as:

$$H(X) = \sum \log p_i.$$

2.6.3.4 Comparison of Entropy Measures

Informal experiments showed that Shannon and Renyi entropy were approximately equally effective as Model Selection Measures, with Burg entropy being slightly less effective. Between Shannon and Renyi entropy, Renyi entropy was chosen for its computational simplicity.

Further informal experiments showed that higher values of α made the measure very slightly more effective; however, the difference was not sufficient to justify the increased computational complexity, thus we chose to use Renyi Entropy with $\alpha = 2$.

2.7 The Measure

2.7.1 Not Quite Entropy

Before continuing, we will more carefully examine the function we have chosen, Renyi Entropy. The definition is:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \left(\sum_{i=1}^{n} p_i^\alpha \right).$$

Because we have decided to use $\alpha = 2$, we can simplify the definition:

$$H_2(X) = -\log\left(\sum_{i=1}^{n} p_i^2\right).$$

Because the log function is monotonic increasing and we are only interested in "greater than" and "less than" relationships of the measure between models, not the actual values, we can remove the log, leaving us with a new function related to Renyi entropy:

$$H_{new}(X) = -\sum_{i=1}^{n} p_i^2.$$

Finally, because we wish the measure to be greater for smaller entropies we negate the measure. At this point the measure cannot be reasonably referred to as entropy any more.

$$F(X) = \sum_{i=1}^{n} p_i^2.$$

2.7.2 The Function

Given a data set, X, and a model, specified by a value of k, the measure is defined as follows:

Taking the set of all possible k–partitionings of the data set:

$$\mathcal{P}(X,\, k) = \{\mathcal{C}_1,\, \mathcal{C}_2,\, \ldots,\, \mathcal{C}_m\}$$

and given the k–means cost function:

$$R(\mathcal{C}) = \sum_{c \in \mathcal{C}} \sum_{x \in c} ||x - \overline{c}||^2$$

we define the normalized value of each partitioning as:

$$\mathcal{V}(\mathcal{C},\, \mathcal{P}) = \frac{R(\mathcal{C})^{-1}}{\sum_{\mathcal{C} \in \mathcal{P}(X,k)} R(\mathcal{C})^{-1}}.$$

Now, our measure can be defined:

$$\mathcal{M}(X,\, k) = \sum_{\mathcal{C} \in \mathcal{P}(X,k)} \left(\mathcal{V}(\mathcal{C},\, \mathcal{P}(X,\, k))\right)^2.$$

This measure allows us to determine the Model Fit Value of a given number of clusters, k, for a given data set, X. However, the number of partitionings possible, and thus the

cardinality of $\mathcal{P}(X, k)$, will increase very quickly with k. The function we use is extremely sensitive to this cardinality and so the Model Fit Value will typically decrease with k regardless of the data set. We have not found any way to normalize the Model Fit Value for the number of partitionings but we will consider in the implementation section below how we can make this Model Fit Value useful.

2.8 The Implementation

The measure, as described in the previous section, has some problems that must be addressed for a practical implementation to be developed. The definition of the measure itself can be implemented directly, but we must make some adjustments to the set of partitionings, $\mathcal{P}(X, k)$, that we use. There are two reasons why this is necessary:

1. The space of the possible partitionings grows extremely large, as in the equation below. For implementation to be practical, only a sample of the partitionings can be used.

2. Currently, the Model Fit Value will decrease with k. If we wish to use the measure to choose the correct value of k for a data set, this must be corrected.

If we wish to determine the correct value of k for a data set X, will will consider a set of prospective k values, k_{test}, and evaluate the measure on $\langle X, k\rangle$ for each value of k in k_{test}. We desire that the Model Fit Value will be greatest for the correct value of k. In order to allow this, we must eliminate the bias from the number of partitionings sampled. The only effective technique we have found is use only a sample of the possible partitionings and to use exactly the same size of sample for each model evaluated.

2.8.1 Number of Partitionings

Because the number of partitionings increases with k, this means that the maximum number of partitionings that can be used is the number of partitionings that exists for the minimal value of k considered. Thus, on a given data set, fewer partitionings can be used to compare $k = 2$, $k = 3$, and $k = 4$ than to compare $k = 3$, $k = 4$, and $k = 5$.

The total number of k-partitionings of n data points can be defined as a recurrence:

$$f(k, n) = \begin{cases} 1 & \text{where } k = 1 \\ \frac{k^n - \sum_{i=1}^{k-1} \frac{k!}{i!} f(k-1, n)}{k!} & \text{otherwise.} \end{cases}$$

To explain briefly, each iteration of the summation is subtracting the cases with i empty partitions and the division by $k!$ is removing the partitionings that are only relabelings of other partitionings.

This number can grow very large, but it turns out that the measure performs very well with only a small sample of the partitionings. We will examine in Chapter 3 exactly how many are necessary.

2.8.2 The Partitions

In our implementation we will choose a number of partitionings to sample such that all models being considered have at least that many partitionings. It must also be considered exactly how the partitionings are to be sampled. We will consider some possible classes of partitionings after defining a Lloyd–step, which is useful in describing the partitioning classes.

2.8.2.1 The Lloyd–step

Lloyd's algorithm is the basis of many methods used for solving the k–means problem. It was first proposed in 1982 in [15] and has been modified and recycled many time since. The algorithm starts with an initial partitioning and iteratively improves it until the solution is 'good enough' or until a fixed point is found. The algorithm is not guaranteed to find the optimal solution, however, and is sensitive to the initial partitioning used. Each iterative step is referred to as a 'Lloyd–step' and consists of finding the centers of mass of each partitions of the input and then creating new partitions based on proximity to these centers. More formally, we define the Lloyd–step on data set X and initial partitioning $\mathcal{C}$, with $|\mathcal{C}| = k$ as:

$$\mathcal{L}(\mathcal{C}) = \left\{ \left\{ x | x \in X \wedge \forall_{i \in [1,k]} \|x - \bar{c}_1\| \leq \|x - \bar{c}_i\| \right\}, \right.$$
$$\left. \left\{ x | x \in X \wedge \forall_{i \in [1,k]} \|x - \bar{c}_2\| \leq \|x - \bar{c}_i\| \right\}, \ldots, \left\{ x | x \in X \wedge \forall_{i \in [1,k]} \|x - \bar{c}_k\| \leq \|x - \bar{c}_i\| \right\} \right\}.$$

Simply stated, the function assigns to each output partition those points closer to the center of the corresponding input partition than to the center of any other input partition. Note that this formalization assumes that all distances are unique; in the event on identical distances, a rule is needed to assign points that are equally close to two centers.

Now, with this definition, we can proceed to define the partitioning classes. Note that each category is a subset of the preceding category.

2.8.2.2 General Partitionings

Any k sets that satisfy the partitioning requirements given in Section 1.3.1 are a general partitioning. It is worth noting that the vast majority of these will have similar (and high) costs.

2.8.2.3 Voronoi Partitionings

Voronoi partitionings are much more satisfying to our intuition of what a clustering should look like. The Voronoi partitionings are those for which there exists a set of k points which generate a Voronoi diagram that induces the partitioning. Equivalently, Voronoi diagrams are those for which the convex hulls of each partition have no pairwise intersection.

These partitions are those that 'look' something like proper clusters. It is still quite possible that they are nowhere near to optimal however.

2.8.2.4 Fixed-point Partitionings

The fixed point partitionings are those that are a fixed point under a Lloyd–step described above. Or equivalently, they are the Voronoi partitionings in which the generating points for the Voronoi diagrams are the centers of mass of the partitions. Informally but perhaps more clearly, the Fixed-point Partitionings are those in which every data point is closer to its partition's center of mass than the center of mass of any other partition.

There are typically very few of these. In the context of the search algorithm perspective on Lloyd's algorithm, these are local optima.

2.8.2.5 Optimal Partitionings

These are the partitionings with the lowest possible cost of all k-partitionings on the given data set. There may be multiple optima with the same cost.

2.8.2.6 Interesting Partitionings

The set of partitionings sampled must somehow characterize the Partitioning Value Distribution. This will generally require that some of the higher–value partitionings be sampled as well as some of the more average partitionings. If we are to consider only a sampling of all partitionings, we must endeavor to ensure that a sufficient range of partitionings are considered in order to generate useful results. It is not sufficient to consider, for example, only the fixed-point partitionings, which typically all have relatively high value. For the degree of peakedness of the distribution to be recognized, the peaks must be recognized, but the lower value partitionings are necessary to give perspective on the peaks.

In general, there is a trade-off between considering all partitionings and considering only a restrictive class of partitionings. If all partitionings are considered, there are a very large number of low–value partitionings and thus it is necessary to sample a very large proportion of the partitionings in order to find enough of the high–value partitionings. On the other hand, we must consider at least some of the lower–value partitionings, so can not be too restrictive in our sampling. We find it most effective to consider the Voronoi partitionings; they have sufficient range of value to well-characterize the distribution and they are sufficiently restrictive that not too large a proportion of them must be sampled.

2.8.3 Sampling Partitionings

We use a sampling algorithm that samples Voronoi partitionings in a fairly uniform way. Simply stated, a set of k centers is chosen uniformly from the minimal axis-aligned box containing the data set and each data point is assigned to a partition corresponding to its

nearest center. The method will not find partitionings for which the Voronoi diagram can only be induced using points outside of the minimal axis-aligned box containing the data, but we have found this method to be quite effective in practice.

When asked to find a specified number, m, of k-partitionings, the system will run as follows:

- initialize PARTITIONINGS, an empty set of partitionings
- repeat while PARTITIONINGS has fewer than M elements in it:
  - generate NEWCENTERS, by choosing K centers uniformly from the minimal hypercube containing the data set
  - create NEWPARTITIONING by assigning each data set element to a partition corresponding to the closest element of NEWCENTERS
  - if NEWPARTITIONING has no empty partitions and is not already in PARTITIONINGS:
    * add NEWPARTITIONING to PARTITIONINGS

2.9 Properties of the Measure Implementation

We will consider here some properties of the measure as it is implemented.

2.9.1 A Stochastic Measure

Because we are considering a random sampling of the possible partitionings, our measure becomes random as well. We must accept the possibility that a bad sample of partitionings will result in an undesirable result. As such, when evaluating our measure, we ask only that there be a high probability over a sufficient partitioning sampling that the measure provides an appropriate result. We will examine this in more detail in Chapter 3.

2.9.2 A New Class

It is important to recognize that, in general, the majority of other methods proposed have been, explicitly or implicitly, formalizations of the cost versus k heuristic described in Section 1.5.1, with stability–based methods being the one major exception. The proposed method belongs to an entirely new class, in that it does not examine the cost versus k tradeoff and does not depend on stability.

2.9.3 Scale Invariance

It is easy to show that the measure is scale invariant. We know that given a data set, X, and a second data set, αX, that has been uniformly scaled by constant α, the k-means cost of a particular partitioning of αX will be α^2 times the cost of the equivalent partitioning in X. Also, we must assume that our method for sampling partitionings is scale invariant. It is easy to see that the method proposed above is.

- We now consider our data sets: the original, X, and the scaled version αX
- We take the corresponding sample of partitionings from each $\mathcal{P}(X,\, k)$ and $\mathcal{P}(\alpha X,\, k)$
- We now wish to find the values of the partitionings for each set of partitionings:
  - For $\mathcal{P}(X,\, k)$, the values are:
$$\mathcal{V}(\mathcal{C},\, \mathcal{P}) = \frac{R(\mathcal{C})^{-1}}{\sum_{\mathcal{C}\in\mathcal{P}(X,k)} R(\mathcal{C})^{-1}}.$$
  - For $\mathcal{P}(\alpha X,\, k)$, the values are, by the scaling properties of the k–means cost function:
$$\mathcal{V}(\mathcal{C},\, \mathcal{P}) = \frac{\alpha^2 R(\mathcal{C})^{-1}}{\sum_{\mathcal{C}\in\mathcal{P}(X,k)} \alpha^2 R(\mathcal{C})^{-1}}.$$
  - It is now simple to factor α^2 out of the summation in the denominator giving us:
$$\mathcal{V}(\mathcal{C},\, \mathcal{P}) = \frac{\alpha^2 R(\mathcal{C})^{-1}}{\alpha^2 \sum_{\mathcal{C}\in\mathcal{P}(X,k)} R(\mathcal{C})^{-1}}.$$
  - Then the α^2 terms in the numerator and denominator cancel out, leaving us:
$$\mathcal{V}(\mathcal{C},\, \mathcal{P}) = \frac{R(\mathcal{C})^{-1}}{\sum_{\mathcal{C}\in\mathcal{P}(X,k)} R(\mathcal{C})^{-1}}.$$
  - which is exactly the definition of the Values for the non-scaled data set; therefore, the measure is scale invariant.

2.9.4 Multiple Solutions

By the very nature of the measure it is entirely possible to find multiple local optima as discussed in Section 1.4.2. This will also be demonstrated experimentally in Chapter 3.

2.9.5 Other Properties

The other desirable properties we consider all relate to choosing the value(s) of k that we consider to be correct. Although we cannot demonstrate that our measure will do so in all

possible cases, we perform a set of experiments in the next section to explore the measure's behaviour on various sorts of data.

Chapter 3

Experimental Results

3.1 Experimental Framework

While development work was done under Wolfram Research's Mathematica, all final experiments were run using a Java implementation of the measure in order to reduce the necessary CPU time. The behaviour of the measure was explored using experiments on synthetic data and the performance of the measure was evaluated using real-world data.

3.2 Experimental Data

3.2.1 Synthetic Data

For each experiment on synthetic data, the following general procedure was followed (although there are exceptions as noted):

- All parameters were fixed except one or more experimental parameters.
- For each desired set of values for the experimental parameters:
  - The specified number of trials were performed:
    - A new data set was generated according to the specified model and number of samples.
    - The measure was evaluated on the data set for each value of k within the test range.
    - If the provided, true, value of k was that for which the measure returned the greatest value, the trial was regarded as a success.
  - The proportion of successful trials was returned, along with a 95% confidence interval based on the normal approximation of the binomial distribution.

- The success rates for each set of experimental parameters was plotted.

There are many parameters involved in the synthetic experiments, and it is important to understand them thoroughly. As inputs to the measure, one must specify:

- The data model with which to generate the data set.
- The number of data points to sample.
- The number of times the experiment is to be run (for a tighter confidence interval).
- The range of values of k to be tested and which is to be specified as correct.

In addition, each data model has several parameters. The model used in the majority of the experiments is the Gaussian Circle Model, described below. When different models are used in specific experiments, they are described there.

In general, the space of possible models and parameters is far too large to explore exhaustively. Thus, we focus our experiments by considering questions about the measure we wish to answer and formulating experiments that can answer those questions.

3.2.1.1 Gaussian Circle Model

The Gaussian Circle Model is a uniform mixture of unit-covariance, two-dimensional Gaussians, a form of MDMG model. The means of the Gaussians are equally spaced around the circumference of a circle about the origin.

The model takes two parameters: k, the number of Gaussians, and s, the separation, or straight-line distance, between each Gaussian's mean and the means of its neighbours. See Figure 3.1 for a diagram. In order to create the correct separation, the radius of the circle is set to $r = s \times \frac{1}{2} \csc\left(\frac{\pi}{k}\right)$. By varying the value of s, the difficulty of correctly determining the value of k can be controlled. See Figures 3.10, 3.11, and 3.12 for example data sets sampled from this class of models.

3.2.2 Iris Data

Although the synthetic data is very useful for exploring the behaviour of the measure, it is interesting to see how the measure performs on real-world data. We consider a data set consisting of the physical dimensions of certain parts of a number of Irises (flowers). The data has been used in various computational learning experiments, first in 1936 [8]. The Irises measured for the data set come from three different species of Iris. There are 150 samples each with 4 different measurements, the sepal length and width and the petal length and width. See Figure 3.2 for a plot of the Petal and Sepal areas of the data set. This is a two-dimensional projection of the four-dimensional data, but retains the separation characteristics of the full data set.

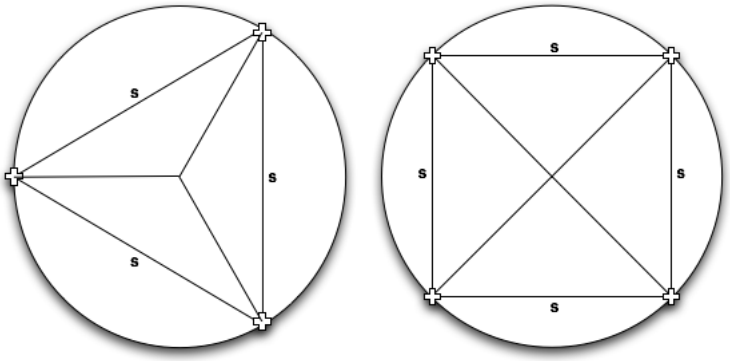

Figure 3.1: A diagram of the placement of the Gaussians' means for the Gaussian Circle Models with $k = 3$ and $k = 4$.

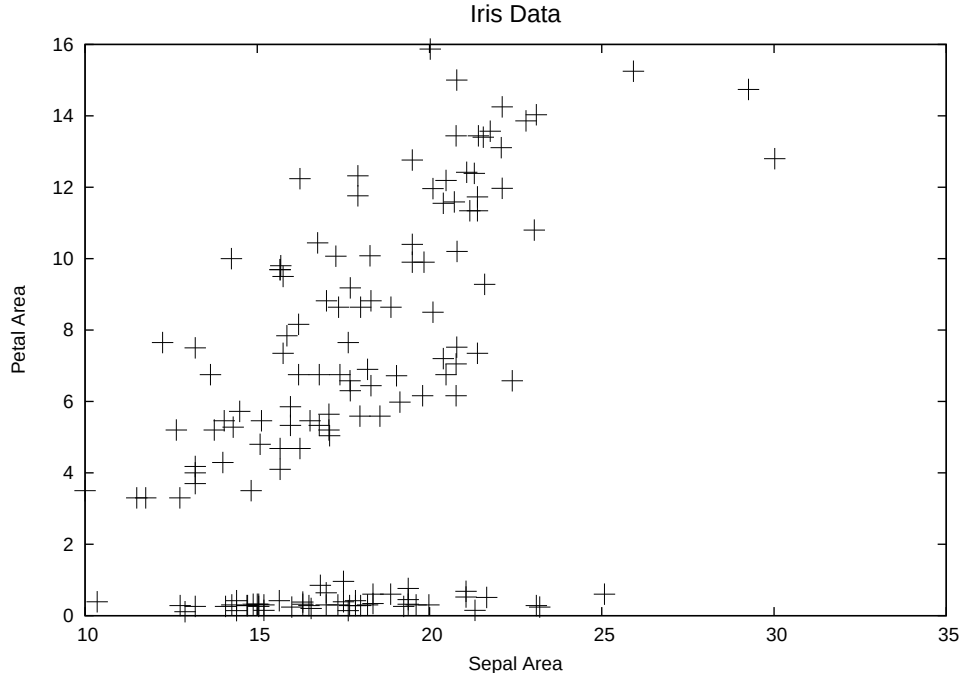

Figure 3.2: Iris data.

There are only two clusters immediately apparent in the data set; however, the larger cluster actually consists of samples from two different species divided at a petal area of around 8. Thus, in consideration of the data source k^* is 3; however, taken out of context, it could be argued that k^* is in fact 2. We will see later (Section 3.3.2) what our measure thinks.

3.2.3 Comparison Data

In order to facilitate a direct comparison of the effectiveness of the measure to other measures in the literature (specifically the Gap Statistic of [11] and the Swiss Stability method of [14]), we test the method on data sets on which the other methods have been evaluated. These data sets are taken from [11, 14]. We consider the Three Cluster Model, with three fairly well separated, but unbalanced Gaussian clusters; the Two Elongated Clusters Model, with two balanced and well-separated clusters made up of points from two lines with a small amount of noise added; and the Five Cluster Model, with five unbalanced, but fairly well-separated clusters.

3.3 Results

3.3.1 Synthetic Data

In these experiments, we use synthetic data to explore the behaviour of the measure.

3.3.1.1 Parameter Reference

We give here a reference of the parameters of the experiments.

- k^* is the number of components in the Gaussian mixture used to generate the data.
- k_{test} is the set of k values that are considered in the experiment. This set is, in a sense, the a priori knowledge about the data.
- n is the number of data points sampled from the distribution.
- m is the number of partitionings sampled.
- s is the separation between the adjacent Gaussians' centers in the data distribution.
- i is the number of times each experiment was run. Numerous runs were used to get better estimates of the probability of success.

3.3.1.2 Finding Sources of Error

We wish to consider what factors might cause the measure to return an undesirable result. Understanding this will better allow us to test the capabilities of the measure. If we assume, for the moment that the correct number of clusters is defined as the number of components in the MDMG model from which the data is sampled, as in the k–defining distributions, we find that the possible sources of failure are:

- m – insufficient samples for the chosen k and n.
- no parameter – measure (incorrectly) defines result other than k^* on this data set.
- n – insufficient data points to distinguish clusters.
- s – insufficient separation to distinguish clusters.

As discussed in Section 1.4.5.5, the last two sources of error above are properties of the data set itself, not the measure. Thus, in these cases, the definition of the correct number of clusters as the number of components in the source distribution is, in fact, incorrect. This is modeled with the error probability function of a k-defining distribution; however, we are not able to actually find explicit values for these functions, so we wish to run an experiment to explore some of these failures. The probability of success was plotted for different values of k^* and n. Other parameters were fixed: $m = 60$, $s = 3$, $i = 1000$, $k_{\text{test}} = [k^* - 1, k^* + 1]$. The separation value was chosen such that the correct answer can be extracted, but the problem is non-trivial.

Examining Figure 3.3, we can notice a few properties. Firstly, in the cases where k is 3 or 4, we see a sharp increase in probability of success with n for the low values of n. This suggests that, in the cases with small values of n, there were insufficient data points. We notice, thereafter, a general decrease in probability of success with n. This can be explained by the fact that m was held constant (and relatively low) while n increased, thus the number of samples became increasingly insufficient. Further, greater values of k^* were progressively less successful, with $k^* = 5$ failing completely. This can, again, be explained due to insufficient samples, as well as an increasingly difficult problem, as separation tolerance is reduced as k increases. We can notice that the lack of samples with n causes a very gradual decrease in effectiveness, whereas, with k^*, the effect is dramatic. With only 60 samples, the measure performed only moderately better than guessing for $k^* = 4$ and even worse for 5.

The question of how many samples are required for successful results leads us to our next experiment.

3.3.1.3 Finding Number of Samples Required

In order to experiment further with the measure, it is useful to have some sense of how many partitioning samples are necessary in order to obtain reliable results. Thus, we performed

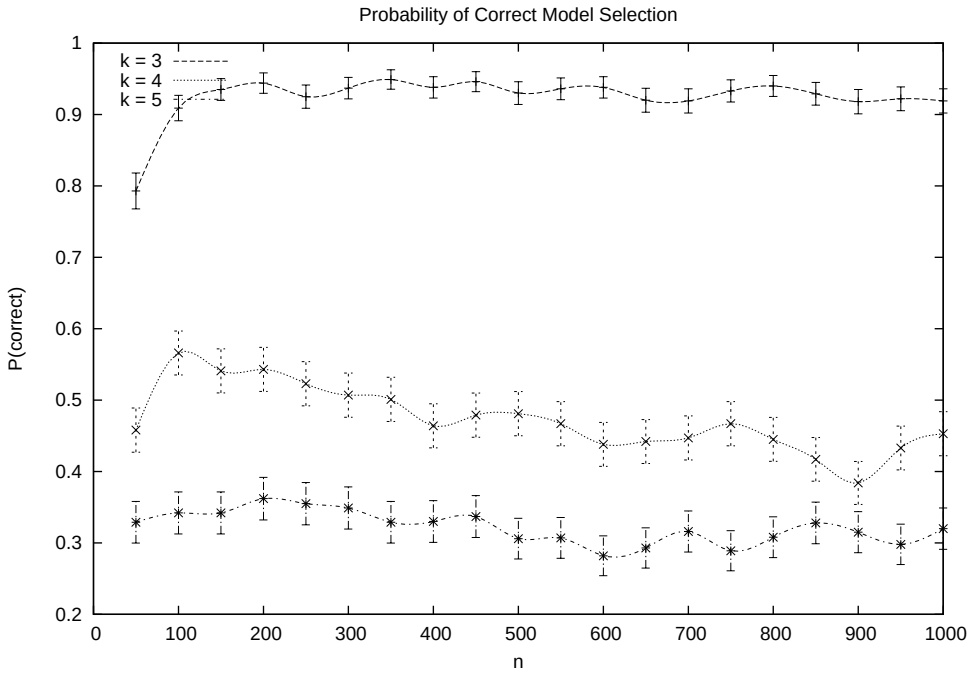

Figure 3.3: Experiment Results. Error bars are 95% confidence intervals.

experiments to determine the number of samples required in order to obtain a 90% success rate for a variety of data sets sampled from Gaussian Circle Models. In all experiments, the fixed parameters were: $i = 500$ and $k_{\text{test}} = [k^* - 1, k^* + 1]$. The number of data points, number of partitioning samples, true number of clusters, and separation were all varied. The results are given in Figures 3.4, 3.5, and 3.6. We find that, independent of the number of data points, the number of samples necessary increases dramatically both as the true number of clusters increases and as the separation between them decreases. Generally, the number of data points has less effect on the required number of samples. We find with a very small number of data points, fewer samples are needed, but once the size of the data set reaches approximately 5000, the number of samples necessary levels off. We also find, however, that for $k = 5$, the effect of larger numbers of data samples is more substantial.

In general, however, the difficulty of the problem (k^* and s) has much more effect than the number of data points on the required number of samples.

3.3.1.4 Finding Toleration for Separation

Here we explored the limits of how well separated clusters must be in order to be distinguished by the measure. We fixed $n = 1000 \times k^*$, $k_{\text{test}} = [k^* - 1, k^* + 1]$, and $i = 500$. We then found how the probability of successful selection varies with the separation for various values of m. We attempted to try increasing values of m until no more performance benefits are found; however, especially in the case of $k^* = 5$, it was not possible to explore the entire potential due to computational limitations. It is possible that better results are possible with larger values of m. See the results for 3 clusters in Figure 3.7, 4 clusters in

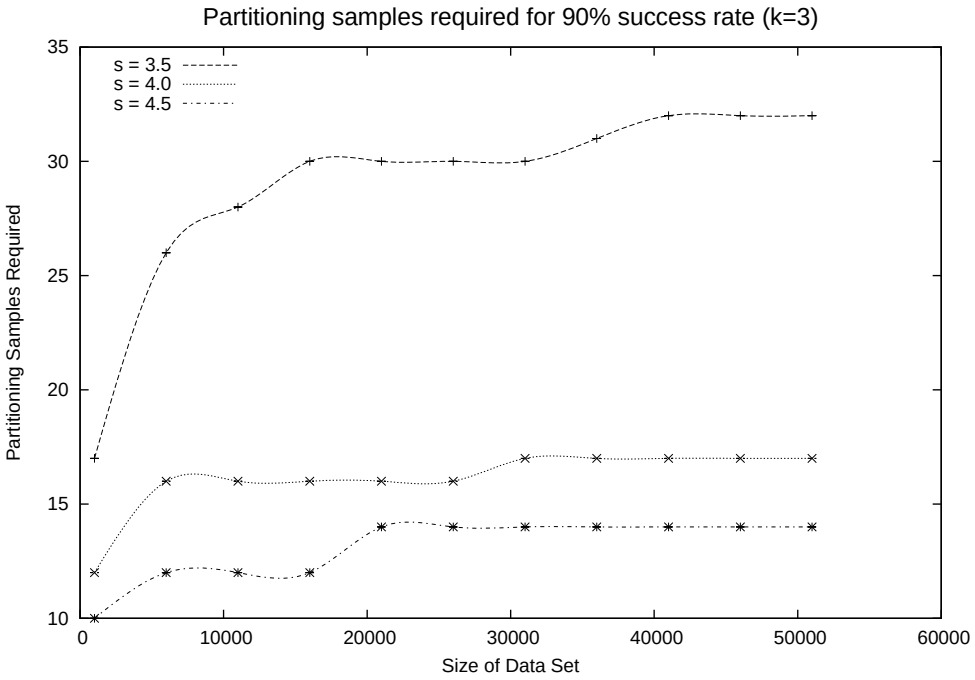

Figure 3.4: Samples required for models with $k = 3$.

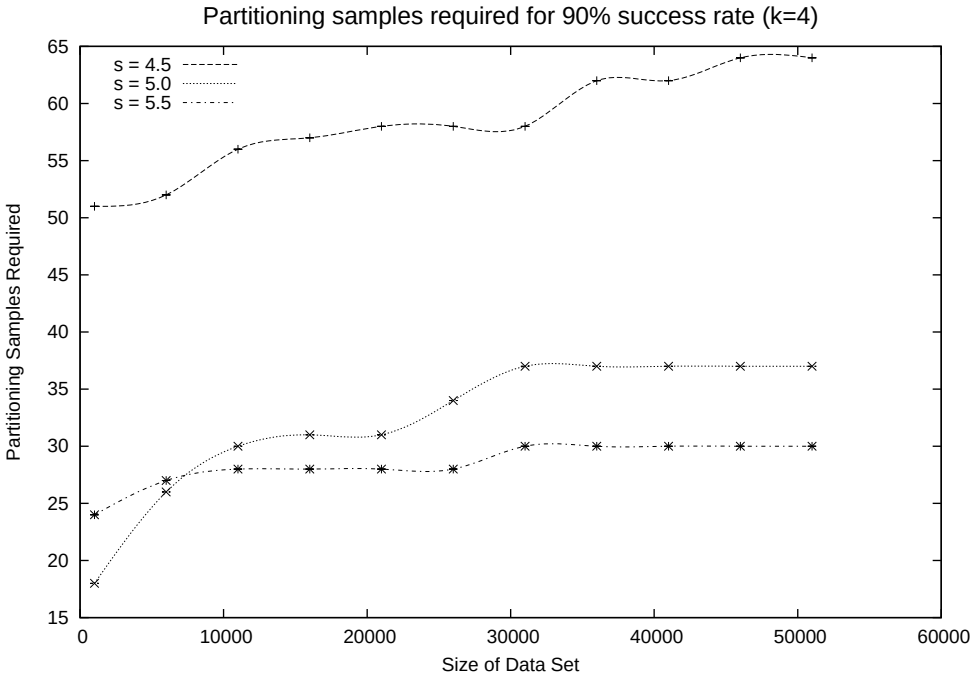

Figure 3.5: Samples required for models with $k = 4$.

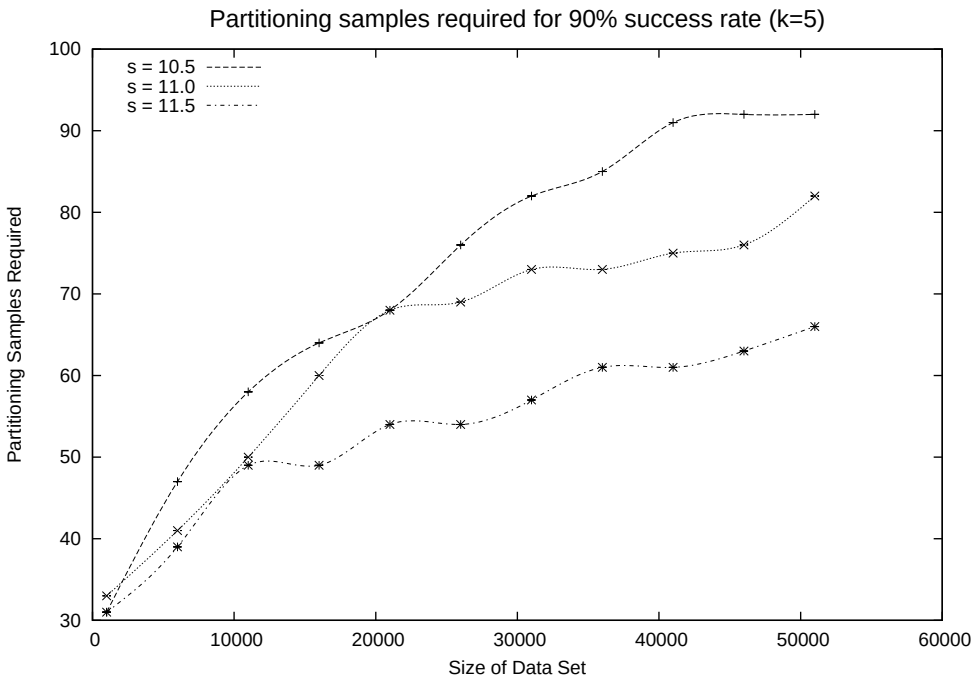

Figure 3.6: Samples required for models with $k = 5$.

Figure 3.8, and 5 clusters in Figure 3.9.

We can see that the minimum required separation for reliable results increases approximately exponentially as the number of clusters increases; examining the data sets in Figures 3.10, 3.11, and 3.12, which show example data sets from the experiment which produce an approximately 90% probability of success, we can see that for $k = 3$ the required separation is very small; however, as k increases, the required separation increases. From a human cognitive standpoint, the $k^* = 5$ data set seem very easy to analyze; however, as in [23], with larger numbers of clusters, the required separations for tractable analysis increases.

3.3.1.5 Finding Effects of Mixed Separation

In previous experiments, there has been a fairly uniform separation between the components' centers. Here we examine the effects of major discrepancies in the separations. To do so, we use a MDMG model with four components arranged at the corners of a rectangle like the one in Figure 3.13. In this data set the horizontal separation between the two pairs of clusters is the 'small' separation and the vertical separation is the 'large' separation.

Depending on the specific values of separation, the value of k^* might be justifiably claimed to be either 2 or 4. For various values of the smaller separation value , we plot the probability of the measure choosing $k = 2$, varying the separation ratio (the value of the larger separation as a factor of the smaller separation).

In Figure 3.14 we give the results, with $m = 50$, $n = 200$, $i = 500$, and $k_{\text{test}} = [2, 4]$ fixed. The experiment was run a second time with $k_{test}\{2, 4\}$ in order to eliminate

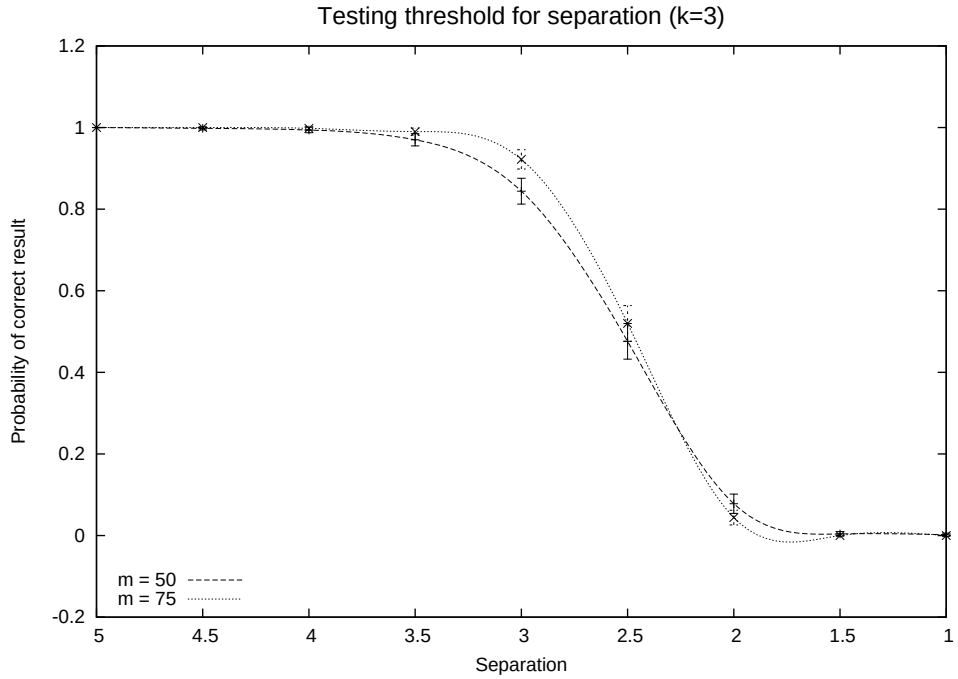

Figure 3.7: Separation tolerance for three cluster model. Error bars are 95% confidence intervals.

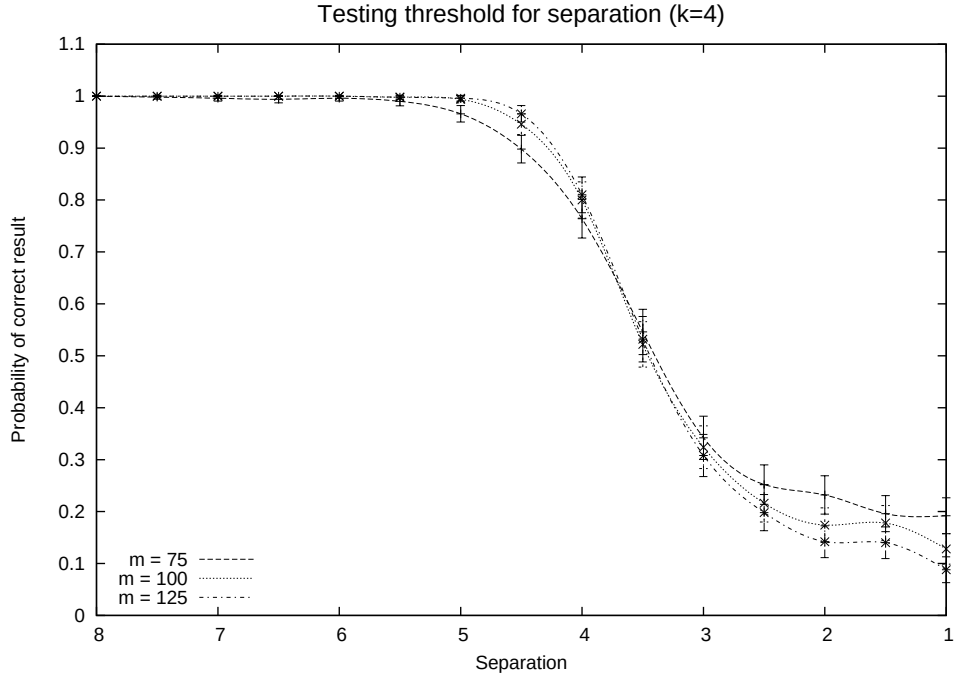

Figure 3.8: Separation tolerance for four cluster model. Error bars are 95% confidence intervals.

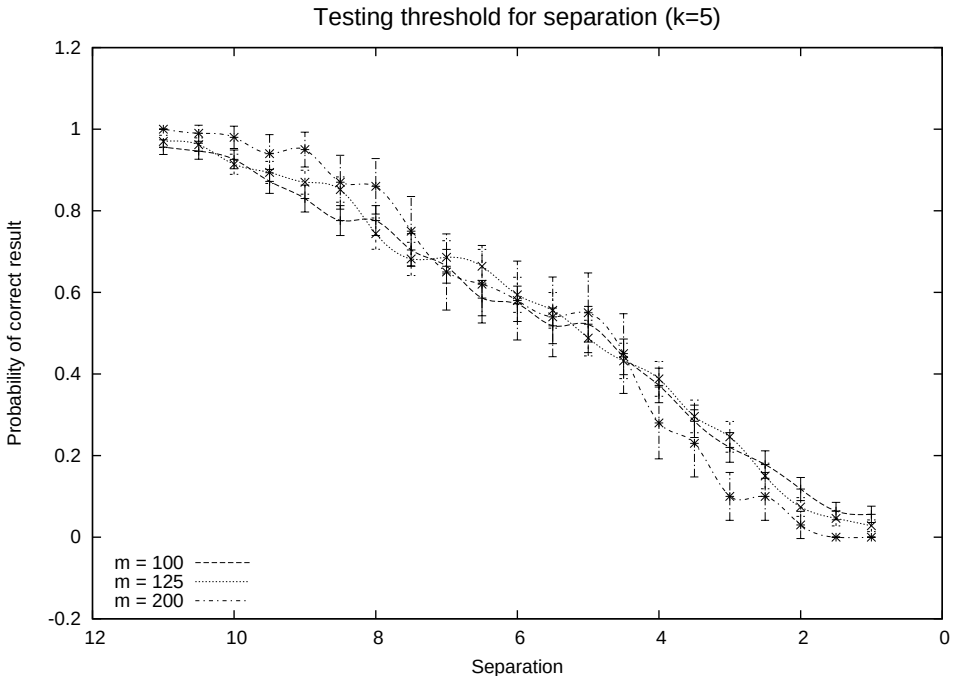

Figure 3.9: Separation tolerance for five cluster model. Error bars are 95% confidence intervals.

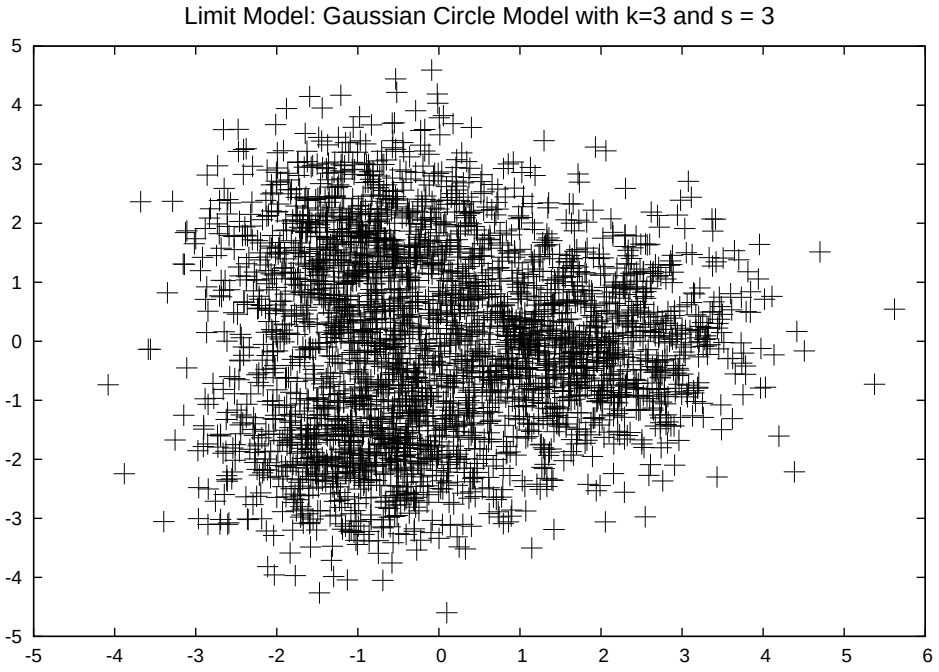

Figure 3.10: Data Set with minimal separation allowing reliable analysis (k=3).

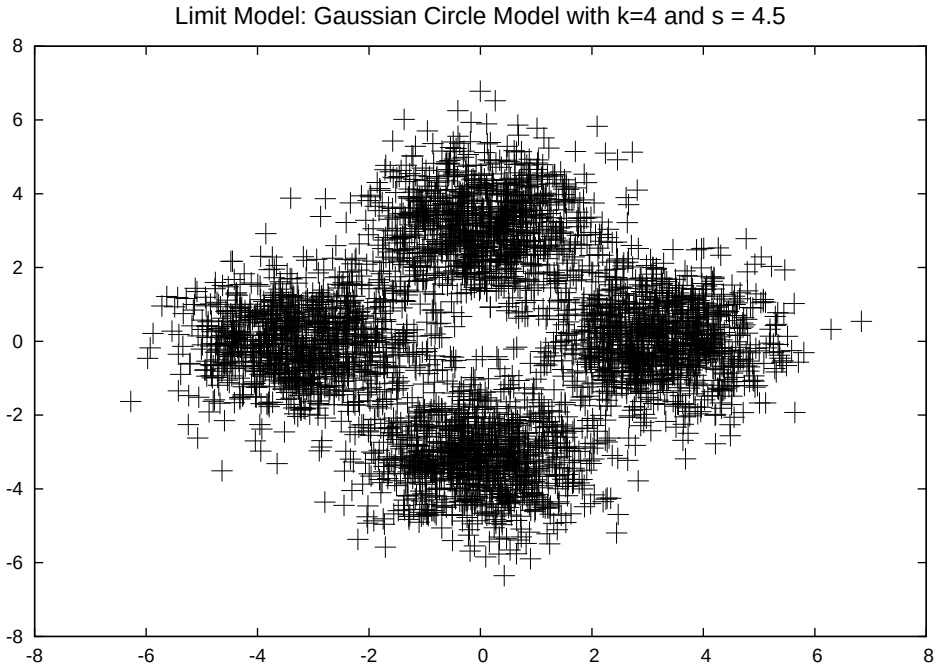

Figure 3.11: Data Set with minimal separation allowing reliable analysis (k=4).

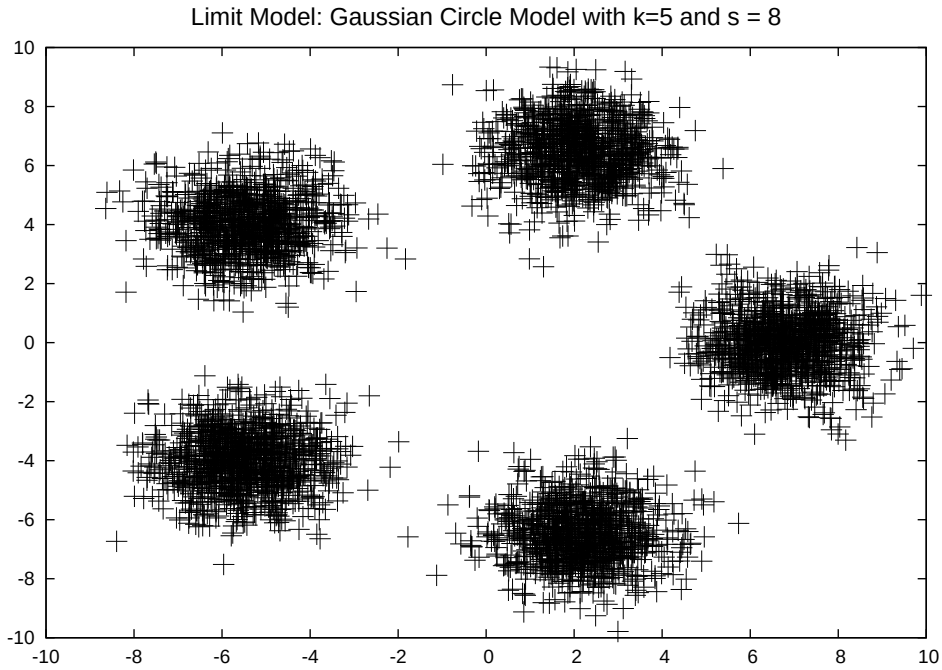

Figure 3.12: Data Set with minimal separation allowing reliable analysis (k=5).

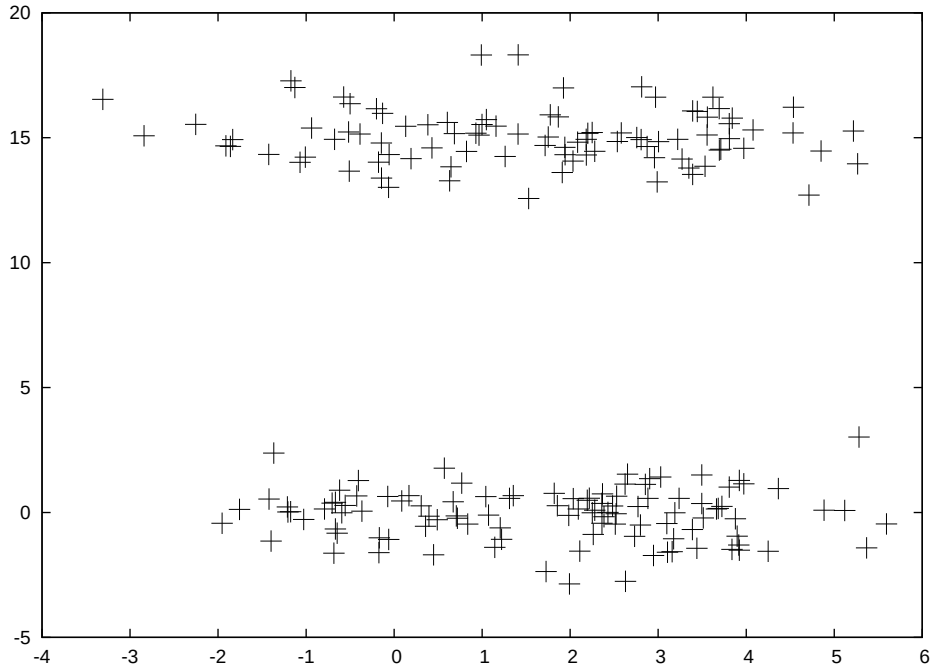

Figure 3.13: Mixed Separation data (4 components) with spacings of 3.0 and 15.0.

interference from cases where $k = 3$ might be selected. The results of the two experiments are almost identical, showing the the measure chose $k = 3$ only very infrequently.

As expected for very small separation ratios, the measure will tend to choose $k = 2$, whereas for larger separation ratios the measure will tend towards $k = 4$. Also, unsurprisingly, smaller values of the short separation will increase the tendency towards $k = 2$ as the close clusters overlap more, eliminating the cluster structure. What is strange is the behaviour when the small separation is equal to 1. The tendency towards $k = 2$ decreases; based on the findings of Section 3.3.1.6 below, we hypothesize that, in this case, the separation is too small resulting in a loss of clustering structure. In this situation, as we will see, the measure will develop a bias towards larger numbers of clusters.

3.3.1.6 Behaviour With Lack of Structure

Here we ask how the measure will behave on a data set without any clustering structure. In this case $k^* = 1$; however, the measure is unable to test this case. Section 4.2 discusses a possible enhancement that might allow this, but for now, we will test models $k = 2$, 3, 4, and, 5. We tested two data sets, a single, unit covariance Gaussian, and a uniform distribution on a unit square. Fixed parameters are $n = 200$, $i = 500$, $k_{\text{test}} = [2, 5]$. The number of partitionings sampled was varied. See Figure 3.16 for the results on the Gaussian distribution and Figure 3.17 for the results on the uniform distribution.

We can see clearly, that in the absence of real clustering structure, the measure will tend to select larger numbers of clusters.

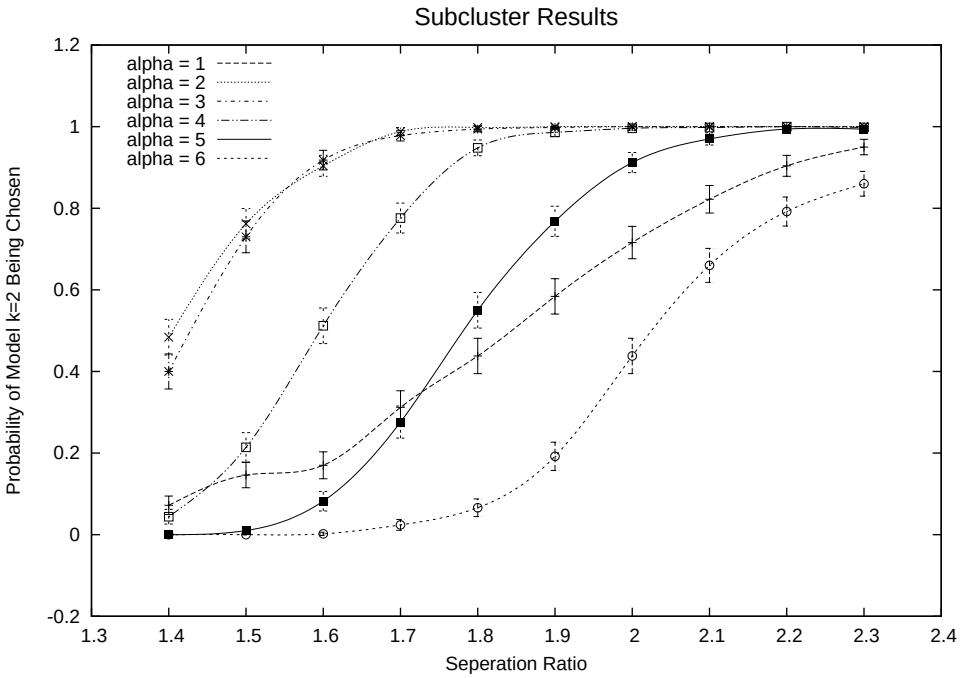

Figure 3.14: Mixed Separation results consider 2, 3, and 4 as possible k values. Error bars are 95% confidence intervals.

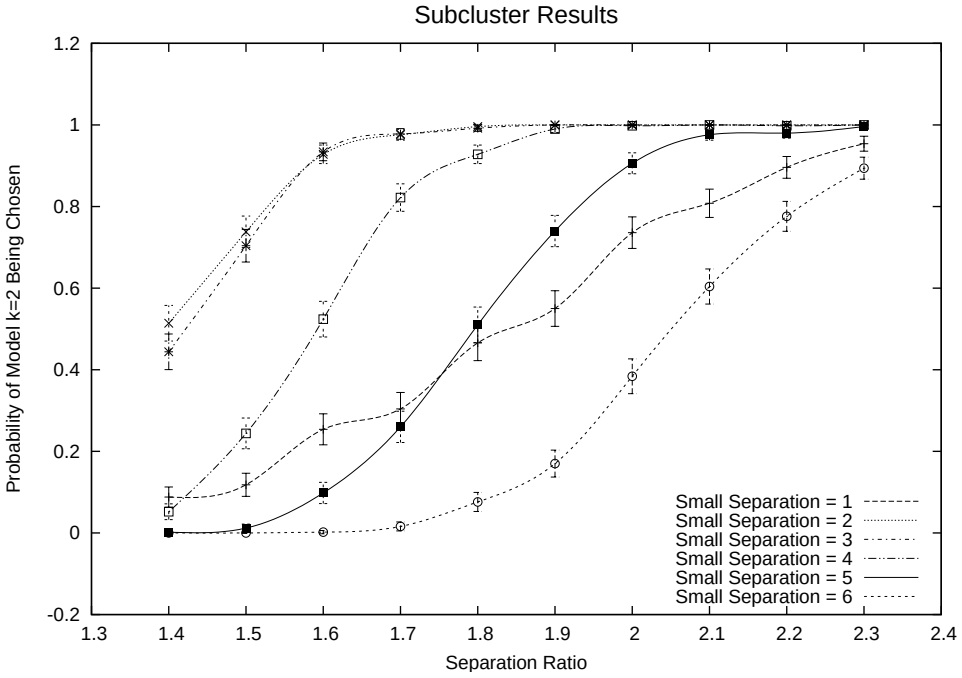

Figure 3.15: Mixed Separation results consider 2 and 4 as possible k values. Error bars are 95% confidence intervals.

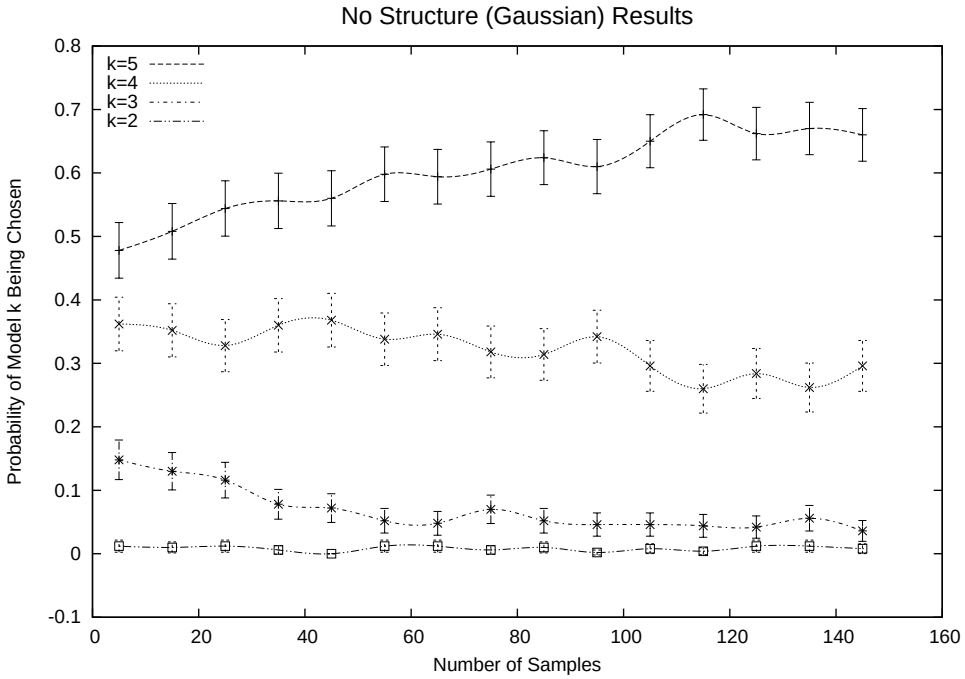

Figure 3.16: Structure Free results on a Gaussian Error bars are 95% confidence intervals.

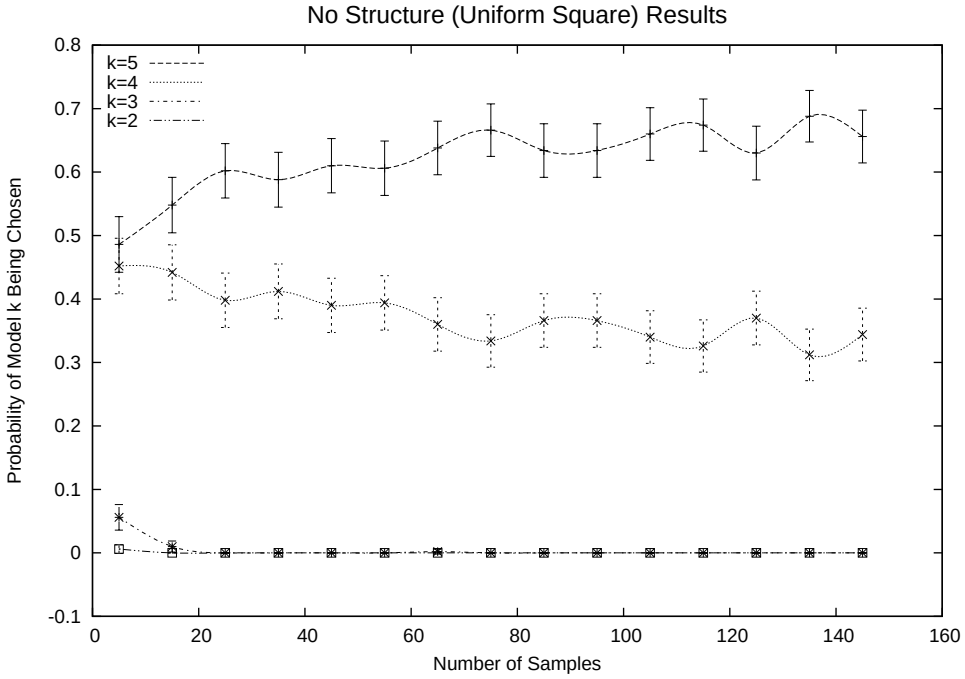

Figure 3.17: Structure free results on a uniform distribution. Error bars are 95% confidence intervals.

3.3.1.7 Finding Bias of Mis-selection

As seen above, in the absence of structure, the measure is biased towards larger numbers of clusters. It was also noted in the experiments of Section 3.3.1.4 that a similar bias existed as the separation became too small. This is unsurprising, as with decreasing separation, the model approaches the case of having no clustering structure.

It remains to ask if a bias exists in the cases where the data set is easily classifiable in general, but an insufficient number of samples are used. To test this case, we ran an experiment on a Gaussian Circle Model with $k^* = 3$, $n = 3000$, $s = 6$, $i = 500$, and $k_{test} = [2, 5]$. This is a fairly easy problem to solve. Remarkably, even with only two partitioning space samples, the minimum with which the measure can operate, the correct model was still chosen in the majority of cases. The results were as follows:

$k = 2$	1.4% of cases
$k = 3$	58% of cases
$k = 4$	27.6%of cases
$k = 5$	13% of cases

Clearly, the bias of failure is towards larger values of k; however, even for such an easy problem, it is very difficult to cause failure.

3.3.1.8 Effects of Uniqueness of Optimal Solution

In order to determine if the measure is effected by the presence or absence of a unique optimal solution, we considered a data set consisting of 100 samples from a mixture of two unit covariance Gaussians with a separation of 6, combined with a single data point between the clusters was considered. In this case, the solution $k^* = 2$ seems reasonable but is in fact an unstable solution as discussed in Section 1.5.2. With fixed parameters $m = 50$, $n = 101$, $s = 6$, $i = 100$, and $k_{test} = [2, 6]$, the measure chose $k = 2$ for every trial. Clearly the absence of a unique optimum is not an important factor for the measure.

3.3.1.9 Effects of Unbalanced Clusters / Outliers

So far, our experiments have been exclusively on uniform mixtures, each cluster being assigned a probabilistically equal proportion of the data points. Here we considered the measure's behaviour on unbalanced clusters using a model with three clusters, one with weight $\frac{1}{3}$, one with weight $(\alpha)\frac{2}{3}$, and one with weight $(1-\alpha)\frac{2}{3}$. The parameter α controlled the degree of balance, with $\alpha = 0.5$ being perfectly balanced and $\alpha \in 0$, 1 being completely unbalanced in either direction. We perform an experiment, varying α and examining the probabilities of the measure choosing either $k = 2$, $k = 3$, or $k = 4$. The fixed parameters are: $k^* = 3$, $n = 3000$, $m = 75$, $i = 500$, and $s = 4.5$. The results are given in Figure 3.18.

It can be seen that, as expected, as long as the clusters are not extremely unbalanced, the measure will choose $k = 3$. Once the clusters are sufficiently unbalanced, the small cluster can be considered a set of outliers rather than a true cluster and the measure

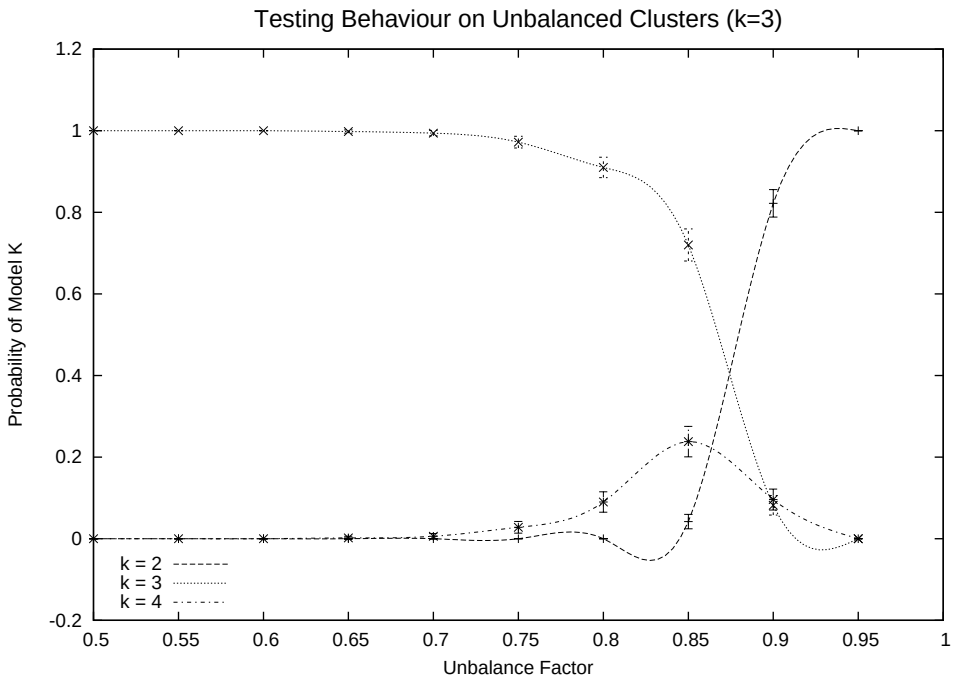

Figure 3.18: Unbalanced cluster results. Error bars are 95% confidence intervals.

switches to $k = 2$. What is interesting, however, is that in the transition regime (around $\alpha = 0.85$), the measure will occasionally select $k = 4$.

3.3.1.10 Effects of Different-sized Clusters

In order to verify that different cluster sizes (different sizes obtained by setting covariance matrix to scalar factor of identity matrix) will not interfere with the measure's operation, experiments were run on a Gaussian Circle Model with three components, two of normal size, and one with size varied from one tenth of normal size to twice normal size. The measure performed as expected, finding $k = 3$ in each case.

3.3.1.11 Effect of Non-circular Clusters

As well as considering clusters of different sizes, it is worthwhile to consider non-circular clusters. For these experiments, a three–component Gaussian Circle Model was used. One cluster was given a covariance matrix of $\begin{bmatrix} 1 & 0 \\ 0 & \alpha \end{bmatrix}$ while the other two used the identity matrix. For the first experiment, the fixed parameters were $k^* = 3$, $n = 2000$, $m = 75$, $i = 500$, and $s = 4$. The value of α was varied and the likelihood of each model was plotted. See Figure 3.19 for the results and Figure 3.20 for an example data set from this model with $\alpha = 4$.

The results are surprising. When the clusters are all relatively circular, the measure selects $k = 3$ as expected; however, as the one cluster becomes elongated, the measure

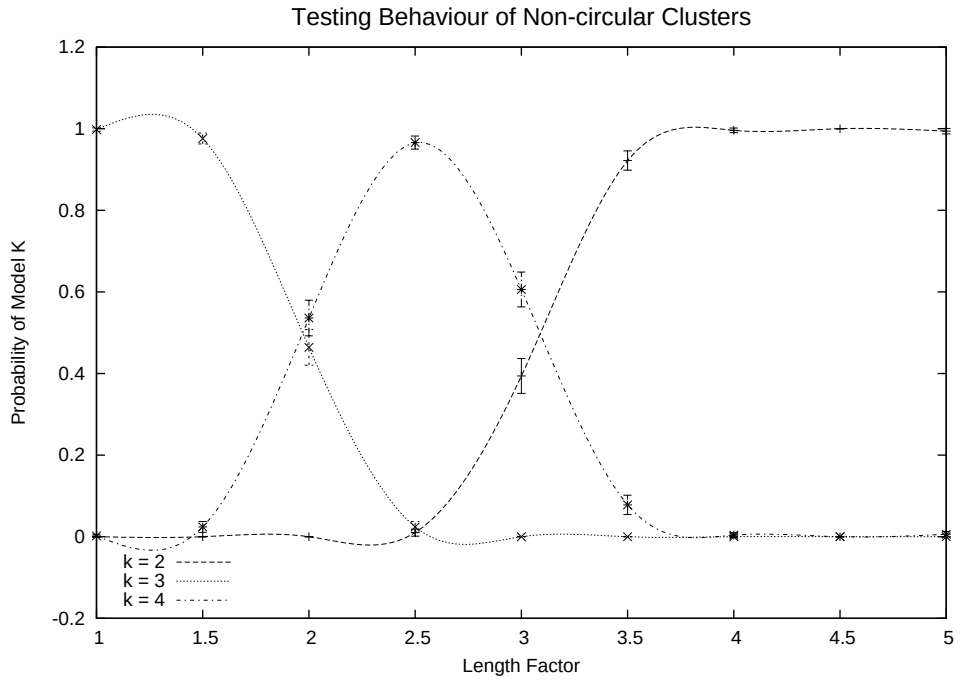

Figure 3.19: Non-circular cluster results. Error bars are 95% confidence intervals.

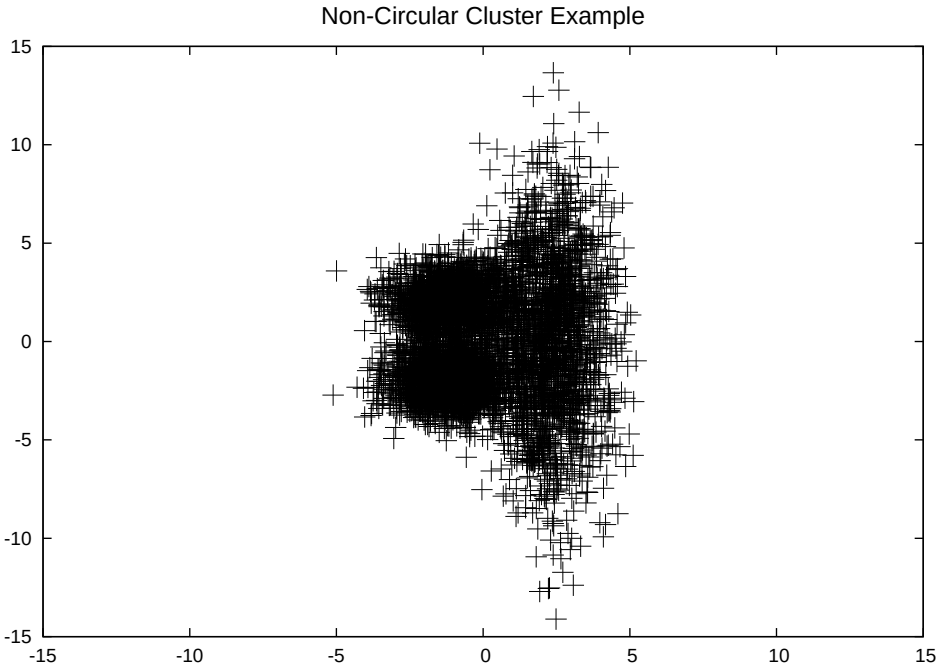

Figure 3.20: Example data from Non-circular Cluster experiment.

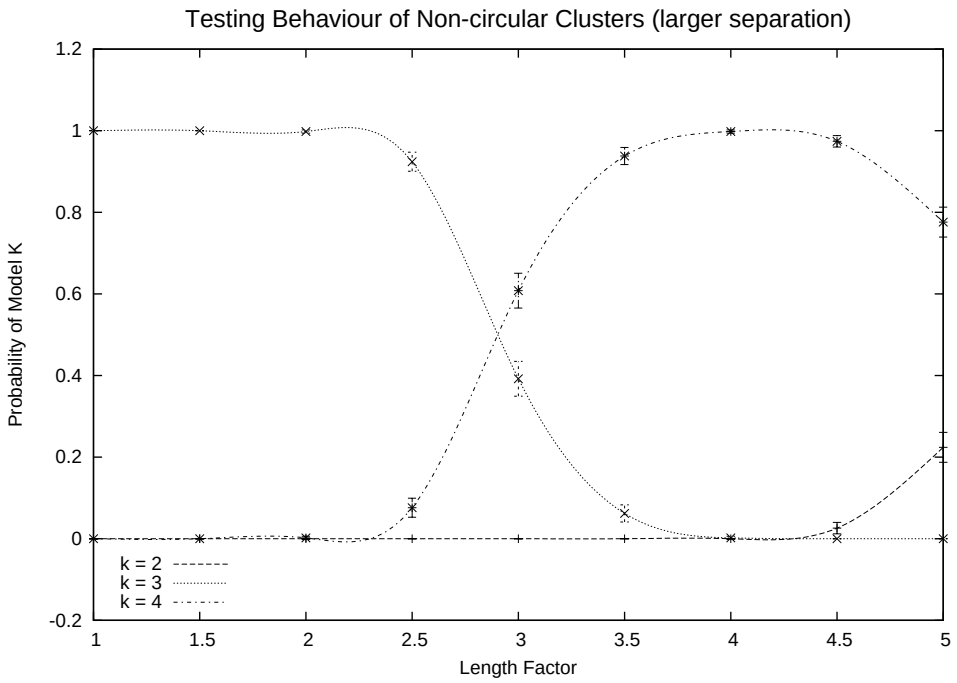

Figure 3.21: Non-circular cluster results with larger separation. Error bars are 95% confidence intervals.

will switch to $k = 2$ with a short regime of $k = 4$ in the middle. It is possible that this middle regime is another example of the boundary–condition confusion evidenced in the experiments in Section 3.3.1.9. The choice of $k = 2$ is more difficult to explain, however. It could be argued that the two circular clusters look like one cluster in the context of the elongated cluster. To explore this hypothesis, we repeated the experiments with greater separation (6). The results are in Figure 3.21.

This time, the measure will tend towards $k = 4$ as the elongation of the cluster increases. It appears it is choosing to divide the longer cluster into two clusters instead of merging the two smaller clusters as in the previous case.

3.3.2 Iris Data

The results for this experiment are especially interesting, although difficult to explain. As discussed in Section 3.2.2, the data set could be argued to have either 2 or 3 clusters: 2 from visual examination, or 3 based on the context of the data. The measure was tested for values of k ranging from 2 to 6 for 1000 iterations each with various numbers of samples. The results are presented in Figure 3.22.

As can be seen in the figure, the measure will select $k = 2$ as the correct model for sufficient samples. This is the model that is correct based on observation of the data; however, for smaller numbers of samples, the measure is more likely to select $k = 3$, the model that is correct based on data context. It is important to note that this is not a

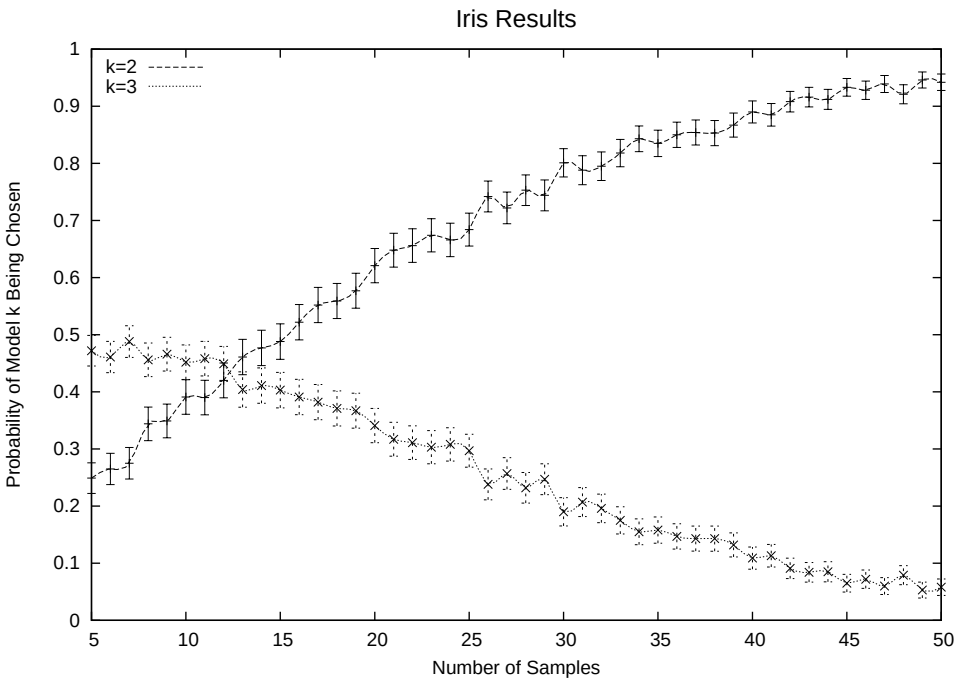

Figure 3.22: Iris Experiment Results. Probabilities do not sum to one due to other models considered. Error bars are 95% confidence intervals.

statistical anomaly as the experiment is easily repeatable. The general bias towards larger k values with smaller numbers of samples found in Section 3.3.1.7 is far too small to explain this behaviour.

3.3.3 Comparison Data

The measure was tested for 50 trials on each of the three comparison data sets, the Three Cluster Model [11], the Two Elongated Clusters Model [11] , and the Five Cluster Model [14]. The results of these trials, as well as published results of trials for the comparison methods discussed in Section 1.5, are summarized in Figure 3.1 below. "48/50 correct " indicates that the given measure determined the correct k value in 48 out of 50 total trials.

- For the Three Cluster Model, I considered {2, 3, 4, 5} as possible k values and used 100 samples for the trials.
- For the Two Elongated Clusters Model, I considered {2, 3, 4} as possible k values and used 10 samples for the trials.
- For the Five Cluster Model, I considered {3, 4, 5, 6} as possible k values and used 20 samples for the trials.
- For the results for the Gap Statistic on the Five Cluster Model, the paper did not specify if the p.c. or uniform method was used. It is assumed that both methods failed.

Data Set	Three Cluster	Two Elongated Clusters	Five Cluster
Correct k	3	2	5
Gap Statistic (p.c.)	48/50 correct	50/50 correct	0/1 correct*
Gap Statistic (unif.)	49/50 correct	0/50 correct	0/1 correct*
Stability	1/1 correct	no result available	1/1 correct
Entropy	49/50 correct	50/50 correct	20/50 correct

Table 3.1: Comparison Data Results

3.3.4 Performance

To test the speed of the measure, we give the time required for a single iteration of each of the limiting case experiments from Section 3.3.1.4. As a reference, the experiment parameters are: $n = 1000 \times k^*$, $k_{test} = [k^* - 1, k^* + 1]$, and $i = 500$ (although resulting times are divided by 500 to give result for a single run). The experiments were performed using the Java implementation under Mac OS X, on a 2GHz Intel Core 2 Duo with 1GB of RAM. A new data set was sampled for each iteration and these times include the data sampling. It should also be noted that three different models were evaluated for each experiment, as k_{test} had three elements in each case. The results are as follows:

$k^* = 3$, $s = 3$, $m = 75$	2.65 seconds per iteration with 94% probability of success
$k^* = 4$, $s = 4.5$, $m = 100$	7.52 seconds per iteration with 94% probability of success
$k^* = 5$, $s = 8$, $m = 200$	64.3 seconds per iteration with 87% probability of success

Note that these are difficult cases, with large data sets, and many typical problems can be reliably solved using fewer samples, allowing for substantially faster processing.

3.4 Summary and Interpretation of Results

The experiments show that the measure generally behaves as desired on the k-defining distributions and typically agrees with human clustering. It is effective with a relatively small number of samples for cases with up to 4 clusters, but begins to require excessive samples (and, thus, computer time) for more clusters. In cases where there is no clustering structure or in some borderline cases in which two possible solutions are balanced, the measure will tend to show a bias towards larger numbers of clusters, but this occurs only in limited cases.

The measure can handle outliers, variations in separation, variations in cluster size, and non-unique optimal solutions robustly. Only in the case of non-circular clusters does some strange behaviour begin to emerge.

3.4.1 Comparison of Methods

We compare our Entropy Method to stability based measures, represented by the Swiss Stability Method and to cost versus k based measures, represented by the Gap Statistic

and find the following:

- All three measures are scale invariant. We demonstrate the scale invariance of the Entropy Method in Section 2.9.3 and the scale invariance of the other methods follows easily along a similar argument.
- Both our Entropy Method and the Swiss Stability Method are capable of handling multiple solutions; the Gap Statistic is not.
- All three methods perform well on samples of extremely well-separated mixtures of Gaussians, which typically have perfect clusterings.
- The Entropy Method will perform well on most k–defining distributions, but sometime behaves incorrectly with non-spherical clusters and cannot efficiently handle large numbers of clusters.
- The Swiss Stability Method will perform well on most k–defining distributions but may behave incorrectly in certain situations where the presence or absence of a unique optimal solution does not correspond with a correct model.
- The gap statistic will perform fairly well on many k–defining distributions, but is not as reliable as the other methods. For example, in [11] it is shown that the gap statistic will occasionally fail on a well-separated mixture of three Gaussians and in [14] it is shown that it will fail on a well-separated mixture of 5 Gaussians.
- The Comparison Data results, summarized in Section 3.3.3, suggests that the Swiss Stability Method is the most accurate, the Entropy Method is as effective in cases with few clusters, but fails with large numbers of clusters, and the Gap Statistic is the least effective.

In conclusion, the Gap Statistic is inferior in terms of both flexibility and correctness. Both the newly proposed Entropy Method and the Swiss Stability Method are generally quite effective; however, the Entropy Method will tend to fail with large numbers of clusters and the Swiss Stability Method may fail in particular cases in which a correct model does not correlate directly with a unique optimal solution. In practical terms, the limitation of the Entropy Method is far more severe than that of the Swiss Stability Method.

Chapter 4

Future Work

4.1 A More General Measure

The proposed measure has been shown to be effective at determining the correct value of k for k–means clustering of a some data sets. There are numerous related problems to which it should be possible to adapt the measure.

First of all, there are many forms of clustering other than k–means, such as path-based clustering [9, 25] or texture-based clustering [26], which is often not formulated as a clustering problem but easily can be. In principle, it should be possible to adapt the measure to work with any form of clustering for which a cost function can be formulated. Further, it should even be possible to select between different forms of clustering; thus, a data set can be evaluated as to whether it would best be clustered with k–means or with, perhaps, a path–based clustering algorithm instead.

Also possible is the development of the clustering model selection measure into a general clusterability measure. Presently, the measure can only be used to compare different models on a fixed data set; however, it should be possible to use the measure on various data sets under a fixed model in order to assess their relative clusterability. Initial experiments suggest that this is quite reasonable; however, it would be necessary to somehow normalize certain properties of the data sets as, presently, certain properties have the effect of skewing the measure undesirably.

4.2 A More Informative Measure

Presently, the measure will indicate which of the models under consideration best fits the data set. If the correct model is completely unknown, it can require the evaluation of the measure on a large number of models to find the correct choice. However, as we saw in Section 2.5, the behaviour of the Partitioning Value Distribution is different when k is underestimated as compared to when it is overestimated. If this distinction could be detected algorithmically, the measure could provide an indication of whether the model

under consideration has too many or too few partitions in it. Even with this information, it might be necessary to examine a fairly large number of models; though, in typical cases, this information would allow very rapid apprehension of the correct value of k.

Additionally, either a characterization of the Partitioning Value Distribution in the case where the data has no clustering structure or a method similar to that used in the Gap Statistic described in Section 1.5.1.1 might allow the measure to determine if a data set has no clustering structure, i.e. $k = 1$.

4.3 A More Powerful Measure

As discussed in Section 2.6.1, only the costs of the sampled partitionings are used in calculation of the measure. A substantial amount of information is thrown away in the form of the partitionings themselves. Given a data set and model with numerous near-optimal partitionings, there is a substantial difference between the case in which each of these partitionings is quite distinct and the case in which they are all very similar, differing only in the classification of a few of data points.

It is possible that the incorporation of this additional information could make the measure much more reliable. As well, it is possible that this information would be necessary to properly generalize the measure as described in Section 4.1 above.

4.4 Understanding Partitionings

As discussed in Section 1.3.2.5, it is not possible to prove the correctness of the measure as there is no objectively correct answer to the clustering model selection problem; however, it may be possible to make the measure easier to justify. Presently, the sampling of a fixed number of partitionings is perhaps the greatest weakness in the method's theoretical appeal.

It would be desirable to find some sort of normalization on the number of partitionings in a model so that it would be possible to use all possible partitionings in the measure. This would provide a somewhat justifiable 'true' value that smaller samplings could be an approximation of.

Further, the sampling process could be better understood. The sampling method used is somewhat arbitrary, and the measure is certainly sensitive to the particular sampling method chosen. A better understanding of the sampling process could lead to an improved, or at least more justifiable, measure. It is also possible that a better sampling process could be found which might improve the performance of the measure.

4.5 A Stronger Justification

The measure has some very nice properties that could potentially make it a reasonable candidate to define the correct number of clusters in any given data set. The relationship to entropy allows the measure to be interpreted as determining the actual degree of orderedness of a given data set under a given model, which gets right to the heart of the structure that clustering searches for. However, the actual connection to entropy is somewhat tenuous. With further analysis, it might be possible to strengthen the justification of the measure, allowing it to be proposed as a 'solution' to the k problem.

Chapter 5

Conclusions

The problem of determining the number of clusters in a data set is highly ambiguous in many cases and can be difficult to solve even when there is a clear solution. Nonetheless, the problem is important for the effective use of clustering in many fields, and is also important from a theoretical perspective, as a thorough understanding of how to determine the number of clusters in a data set will necessarily provide a thorough understanding of the structure that clustering is intended to uncover.

Past approaches to the problem have either relied on the rate of decrease of the cost of the optimal solution as k increases, which overly simplifies the problem, or relied on stability, which in not a sufficient criterion. The measure described here provides a novel approach based on the actual information content of the clustering structure, which provides a natural argument for the correctness of the measure; experiments showed the measure, although not yet generally effective, has great promise.

With further development, the measure might be extended to be a much more general measure of clusterability which could lead to a deeper understanding of what clustering really is.

Appendix A

Glossary

Terminology:

- The terms 'cluster' and 'partition' are used interchangeably and represent a subset of a data set. In some cases 'set' is also used as a synonym.
- The terms 'clustering' and 'partitioning' are used interchangeably and represent the division of a data set into several non-intersecting clusters or partitions.

Mathematical Notation:

- x is a data point, generally a real-valued vector, $x \in \mathbb{R}^d$.
- d is the dimensionality of the data.
- $||\cdot||$ is the Euclidean Distance.
- X is a set of data points, or a data set, $X = \{x_1, x_2, \ldots, x_n\}$.
- n is the cardinality of a data set $n = |X|$.
- 2^X is the power set of a data set.
- $\mathcal{X}$ is the set of all possible data sets. The dependence on the domain is implicit.
- c is a partition of a data set, $c \in 2^X$.
- $\overline{c}$ is the mean of a partition.
- $\mathcal{C}$ is a partitioning of a data set, $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$.
- k is the number of clusters is a clustering or the cardinality of a partitioning, $k = |\mathcal{C}|$.
- $k^*(X)$ is the set of 'correct' numbers of clusters in data set X.

- $\mathcal{P}$ is a set of partitionings, $\mathcal{P} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$.
- m is the cardinality of a set of partitionings, $m = |\mathcal{P}|$.
- $R(\mathcal{C}, k)$ is the cost of partitioning $\mathcal{C}$ under the k–means model. Note that the k parameter is redundant but included for clarity.
- $V(\mathcal{C}, k)$ is the value of partitioning $\mathcal{C}$ under the k–means model. Note that the k parameter is redundant but included for clarity.
- $\mathcal{V}$ is a set of partitioning values, $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$.
- v is a partitioning value, $v \in \mathbb{R}$.
- $\mathcal{M}(X, k)$ is the model selection measure for the k–means model on data set X.
- $\mathcal{M}(\mathcal{P})$ is the model selection measure for the k–means model on data set X, using the set of partitionings $\mathcal{P}$. Dependence on X and k is implicit through $\mathcal{P}$.

List of References

[1] Shai Ben-David, Dávid Pál, and Hans Ulrich Simon. Stability of k-means clustering. *Proceedings of COLT 2007*, Aug 2007.

[2] Shai Ben-David, Ulrike von Luxburg, and Dávid Pál. A sober look at clustering stability. *Proceedings of COLT 2006*, pages 415 – 426, Mar 2006. 14

[3] Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing Pacific Symposium on Biocomputing*, pages 6–17, Jan 2002. Use of stability for determining correct k. Some measures of partitioning similarity. Some good references. Quality of a given choice of k is determined by finding distribution of similarity of subsample clusterings. 14

[4] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3(1):1–27, Jan 1974.

[5] A. Casillas, M. T. González de Lena, and R. Martínez. Document clustering into an unknown number of clusters using a genetic algorithm. *Lecture Notes in Computer Science*, 2807/2003:43–49, Feb 2004.

[6] Andrew Coward. The recommendation architecture model for human cognition. *Proceedings of the Conference on Brain Inspired Cognitive Systems, University of Stirling, Scotland.*, 2004. 7

[7] Scott Epter, Mukkai Krishnamoorthy, and Mohammed Zaki. Clusterability detection and initial seed selection in large data sets. *The International Conference on Knowledge Discovery in Databases*, 1999. 7

[8] R. A. Fischer. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, Aug 1936. 30

[9] A.D. Gordon. Classification (2nd edition). 1999. 50

[10] Peter Grassberger. Entropy estimates from insufficient samplings. *Arxiv preprint physics*, Jun 2006.

[11] Trevor Hastie, Robert Tibshirani, and Guenther Walther. Estimating the number of data clusters via the gap statistic. *Journal of the Royal Statistical Society: Series*

B (Statistical Methedology), 63(2):411–423, Aug 2001. I could steal there idea for considering the k=1 case!!! requires finding optimal solutions good experiment setup, use they';re synthetic data ideas basic method error-k curve compared to that of uniform distribution. 13, 14, 32, 47, 49

[12] J Kleinberg. An impossibility theorem for clustering. *Proc. of the 16th conference on Neural Information Processing Systems*, 2002. 6

[13] W. J. Krzanowski and Y. T. Lai. A criterion for determining the number of groups in a data set using sun-of-squares clustering. *Biometrics*, 44(1):23–34, Mar 1988.

[14] Tilman Lange, Volker Roth, Mikio L Braun, and Joachim M Buhmann. Stability-based validation of clustering solutions. *Neural computation*, 16(6):1299–323, Jun 2004. 14, 32, 47, 49

[15] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. 24

[16] Glenn W. Milligan and Martha C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50(2):159–179, Jun 1985.

[17] Ilya Nemenman. Entropy estimation: An overview. *NIPS'03 workshop on Entropy and Information Estimation*, Dec 2003. 20

[18] Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. *Proceedings of 47th annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 165–176, Aug 2006.

[19] Liam Paninski. Estimating entropy and information, discretely. Dec 2003.

[20] Alexander Rakhlin and Andrea Caponnetto. Stability of k-means clustering. *Advances in Neural Information Processing Systems*, 19, Aug 2007.

[21] Alfréd Rényi. On measures of entropy and information. *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, Aug 1960. 16, 21

[22] C E Shannon. The mathematical theory of communication. 1963. *MD computing : computers in medical practice*, 14(4):306–17, Jan 1997. 20

[23] Nathan Srebro, Gregory Shakhnarovich, and Sam Roweis. An investigation of computational and informational limits in gaussian mixture clustering. *ACM International Conference Proceeding Series; Proceedings of the 23rd international conference on Machine Learning*, 148:865–872, Feb 2006. 11, 12, 36

[24] D Steinley. Stability analysis in k-means clustering. *Br J Math Stat Psychol*, Feb 2007.

[25] Werner Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *Journal of Classification*, 2003. 50

[26] K.S. Thyagarajan, Tom Nguyen, and Charles E. Persons. A maximum likelihood approach to texture classification using wavelet transform. *Image Processing - Proceedings. ICIP-94., IEEE International Conference*, Mar 1994. 50