university of groningen / faculty of science and engineering

# Determining k in k-means clustering by exploiting attribute distributions

Bachelor's Project Thesis

Oscar Bocking, O.E.Bocking@student.rug.nl
Supervisor: Dr L. Schomaker

**Abstract:** Methods for estimating the natural number of clusters (k) in a data set traditionally rely on the distance between points. In this project, an alternative was investigated: exploiting the distribution of informative nominal attributes over the clusters with a chi-squared test of independence, to see which value of $k$ partitions the data in a way that is least likely to be random. Artificial data sets are used to assess the strategy's performance and viability in comparison to a well-established distance-based method. Results indicate that the proposed strategy has a tendency to overestimate $k$, and only performs consistently with some types of attribute. Despite this, it has value as a heuristic method when attributes are available due to non-reliance on distance information.

## 1 Introduction

In k-means clustering, the aim is to divide objects into $k$ groups, while minimising the sum of the squared difference between the cluster means and their members. Minimising this value is NP-hard (Drineas, Frieze, Kannan, Vempala, and Vinay, 2004), and so algorithms will generally converge to a local minimum. The technique is versatile, with applications in meteorology (Arroyo, Herrero, and Tricio, 2016), genetics (Tibshirani, Hastie, Eisen, Ross, Botstein, and Brown, 1999), marketing (Punj and Stewart, 1983), and countless other fields; however it requires the number of clusters $k$, to be given. While the method proposed in this thesis could theoretically be applied with any clustering technique where $k$ needs to be given, the focus will be on k-means clustering since it is a simple algorithm, but one of the most widely used (Berkhin, 2006).

### 1.1 K-Means

The standard k-means algorithm, sometimes referred to as Lloyd's algorithm (Lloyd, 1957), is applied as follows:

1. Initialise $k$ cluster centres. The simplest way to do this is to randomly select $k$ datapoints.
2. For each object, assign it to the cluster that has the closest centre. Typically Euclidean distance is used, however other measures of similarity may be preferred.
3. Recalculate the cluster centres as the mean of their members.
4. Repeat steps 2 and 3 until the algorithm converges.

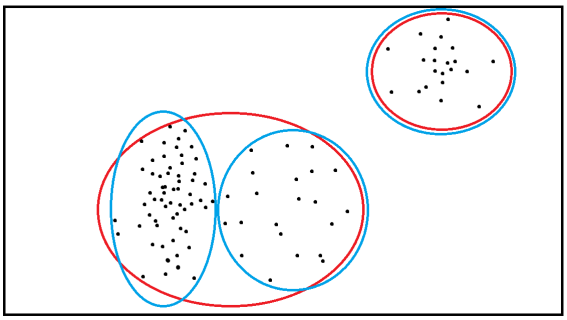

**Figure 1.1: Clustering data with k=2 (red) and k=3 (blue)**

When faced with a cluster-analysis, deciding how many groups are present in the data is a complex problem of its own. In Figure 1.1, different people may interpret the distribution of data as belonging

to 2 or 3 groups, and either division could make sense depending on the context. While a myriad of techniques have been devised to automatically determine $k$, each of these may be more or less suitable depending on the specific characteristics of a problem (Mirkin, 2011).

## 1.2 The Elbow Method

Generally, $k$ is found by clustering with a number of values of $k$, and then the best value can be chosen by some criterion (Jain, 2010). The classic example of this is the 'elbow' method (Thorndike, 1953). For each $k$ the data is clustered, and the average deviation of the points from their cluster means (root-mean-square error) is plotted. Alternatively, the F-statistic can be plotted, which calculates the percentage of the variance that is explained by the partition. In either case, increasing the number of clusters will reduce the error, and so $k$ cannot just be chosen to minimise this value. Instead, the user must look for the elbow, or point of maximum curvature, after which more clusters will provide a smaller improvement. The rationale is that the clustering that explains the majority of the variance with as few clusters as possible is the most natural.

While the theory is logical, in practice finding the elbow can be ambiguous or subjective (Ketchen Jr and Shook, 1996). This is illustrated with an extreme example in Figure 1.2, where there is no discernible elbow.

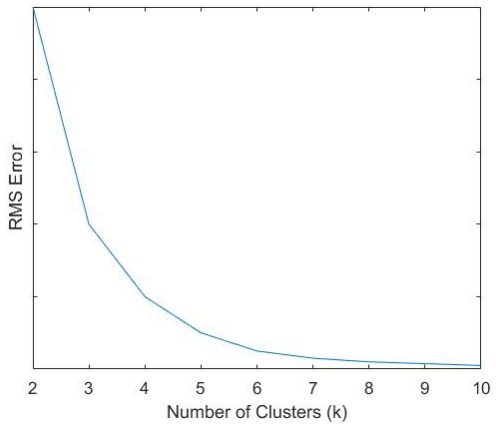


**Figure 1.2: An ambiguous elbow plot**

## 1.3 AIC/BIC

The Akaike information criterion (AIC) (Akaike, 1974) estimates the quantity of information lost in representing data with a model, compared to other models. The main benefit of this over the elbow method is that the calculated figure includes a penalty for the number of clusters used. The model with the lowest AIC is chosen as the best way to cluster the data, and so the selection of $k$ is objective. The Bayesian information criterion (or BIC) (Schwarz et al., 1978) is nearly identical to AIC, however it uses a larger penalty term ($ln(n)k$, where n is the number of data points, rather than $2k$) and so can recommend fewer clusters. BIC tends to be preferred to AIC for clustering problems because it's mathematical formulation is more meaningful in this context (Pelleg, Moore, et al., 2000), whereas AIC is more general.

Both AIC and BIC were designed for more general model selection problems, and more specialised methods tend to be preferred (Mirkin, 2011). There is evidence to suggest that these criteria tend to overestimate the number of clusters in data (Hu and Xu, 2004). Finally, both of these criteria show very little variation when the underlying structure of the data is not well separated (Windham and Cutler, 1992), and so are poorly suited to more difficult problems.

## 1.4 The Silhouette Method

The silhouette method, as described by Rousseeuw (1987), was designed as a method for visualising the suitability of cluster assignments over a data set. For each point, a statistic $s(i)$ between 1 and -1 is calculated based on the average distance to points within the cluster, and those in the closest neighbouring cluster ($s(i) = \frac{b(i)-a(i)}{max(a(i),b(i))}$, where $a(i)$ is the mean distance between i and the rest of its cluster, and $b(i)$ is the lowest mean distance between i and the members of any other cluster). Histograms of the values are presented as in figure 1.3, arranged by cluster and sorted by statistic, giving the 'silhouette'; a visual representation of the suitability of objects to their clusters. While this visualisation can be informative, $\bar{s}(k)$ (the mean silhouette width) can be used as an indication of the overall quality of the partition. Similarly to the previous strategies, $\bar{s}(k)$ can be calculated for a range of val-

ues of $k$, and the partition that maximises this value selected. This method performs well in experiments (Pollard and Van Der Laan, 2002).

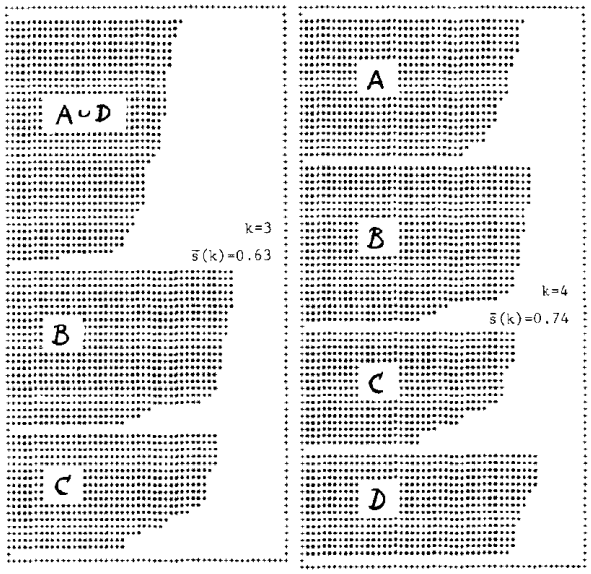


**Figure 1.3: Silhouettes for data clustered when k=3 and k=4. Reprinted from P. J. Rousseeauw's 1987 paper**

Unlike the information criterion approaches, the silhouette method penalises having clusters that are similar to each other, as opposed to the simply the number of clusters. This has the same effect of penalising large values of $k$, but takes into account additional distance information. AIC and BIC don't use such information, since they were designed to be applicable to a range of model selection tasks where this information may not exist. This trade-off between specificity and broadness of application is a recurring theme in clustering: exploiting more information about data *can* yield better results, but the methods will not be applicable to other problems. To achieve the best results, experimenters will often have to transform their data and modify mainstream methods to suit their situation.

## 1.5 The Gap Statistic

The gap statistic proposed by Tibshirani, Walther, and Hastie (2001) is based on the same principle of within cluster variance as the aforementioned elbow method. The sum of the pairwise distances between all points in all clusters is compared to the value that would be expected if the points were distributed evenly. Gap($k$) is the ratio of the logarithms of these two values, and the theoretical optimal number of clusters is that which maximises the gap statistic. In practice, a number of 'expected' distributions are drawn from one reference distribution to be compared with the clustered data, rather than performing a single comparison. The smallest $k$ is chosen for which $Gap(k) \geq Gap(k+1) - sd_{k+1}$. The the standard error being the difference threshold is based on the assumption that a partition that is more resilient to the random perturbations in the expected distributions is a better partition.

One major advantage of the gap statistic is that it can also recommend 1 cluster, essentially suggesting that the data is unsuitable for the clustering algorithm. None of the other algorithms mentioned in this section have such a capability, which can lead to the unfortunate situation where an experimenter sorts data into an unmeaningful partition. The silhouette method for example generally recommends 2 clusters for data that is uniformly distributed (Tibshirani et al., 2001). Because the gap statistic takes data of this kind into account, no preliminary analysis is necessary to decided whether or not the data should be clustered.

## 1.6 Alternatives

There exist a plethora of strategies for determining $k$, and there are also many clustering techniques for which $k$ does not need to be provided; most notably density based clustering methods such as DBSCAN (Ester, Kriegel, Sander, Xu, et al., 1996), and hierarchical clustering methods. However, these alternatives each have their own parameters that need to be determined. In DBSCAN for example, a distance must be determined within which a nearby point is considered a 'neighbour'. Hierarchical clustering generally presents the user with a tree representing the data's structure. This tree can be cut at different positions to give a partition, and so the $k$-decision is essentially deferred rather than avoided.

The premise of this investigation is that $k$ could be chosen based on the distribution of some nominal attribute. When clustering, feature vectors can contain both continuous and nominal attributes. While one might expect some of these attributes to be related to the clusters, categorical variables generally require special accommodation in clustering

(Huang, 1998), which is especially difficult to implement if the impact of the variable is complex or unknown. Additionally, an experimenter may only be interested in groupings that are derived from the numerical data. When clustering the continuous feature vectors, is it possible to make use of additional attribute information to determine $k$?

Firstly, it is assumed that a chosen nominal attribute is expected to have some irregular (and therefore informative) distribution over the final clusters. A better partition of the data will show this relationship more strongly, while a less good partition will show a more random distribution. There may be a $k$ for which the distribution of this attribute is consistently more irregular, indicating that this $k$ provides a more meaningful division of the data, and is therefore the best choice. A chi-squared test of independence could, in theory, be used to assess this. In a pilot experiment (Schomaker, 2017), it was found that a dip in p-value could be observed with this heuristic method (Figure 1.4). The objective of this study is to reproduce this effect in a controlled experiment with artificial data.

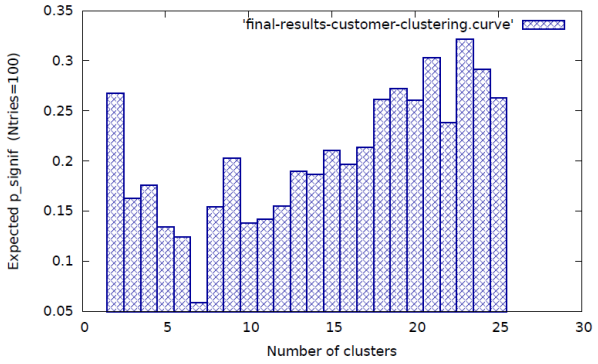


**Figure 1.4: A plot of p-values calculated from chi-squared tests on the distribution of attributes for different values of $k$ in a pilot study (Schomaker, 2017)**

## 1.7 The Chi-Squared test

A chi-squared test of independence evaluates the distribution of counts in a contingency table to approximate the likelihood of the differences between categories being random.

In cryptanalysis a chi-squared test can be used to evaluate whether a message is in natural language (Ganesan and Sherman, 1994; Ryabko, Stognienko, and Shokin, 2004). Comparing the distribution of letters in a message with the expected counts of each letter will approximate the probability of the given distribution belonging to a natural language message. For example, when decoding a Caesar cipher, this frequency analysis can be performed for every possible decryption, and the message identified as that with the lowest $\chi^2$ statistic. The benefit of performing a frequency analysis here is that it allows language to be recognised based on a single characteristic of a complex system; meaning that it can be quickly done by computer.

## 1.8 The Proposal

Fundamentally, the algorithm proposed in this thesis is as follows:

1. Cluster the data set several times for $k = k_{min}, ..., k_{max}$.

2. For each partition, perform a chi-squared test of independence on the distribution of a categorical variable over the clusters, and note the p-values.

3. Create a histogram of the mean p-value for each value of $k$.

4. Choose the $k$ for which the mean p-value is lowest.

Since the k-means algorithm is non-deterministic, clustering several times will allow a user to be more confident in their decision of $k$. An outlier that produces a low p-value may seem like a great partition, but a mediocre partition could show an irregular pattern by chance. *Consistently* lower p-values are more indicative of a useful $k$. Additionally, as discussed with the gap statistic (Tibshirani et al., 2001), resilience to the random variation in the k-means algorithm is an indicator of a good $k$. For these reason, choosing the partition that gave the lowest p-value in a single instance would be a mistake. The mean p-value should therefore be used, since it gives an impression of the overall appropriateness of $k$.

# 2 Method

To test the proposed method, data sets were generated to contain clusters with related attributes. The attributes were compared to see which were most suitable, and then the results were compared with those of the silhouette method on the same data. Extra implementation details not present in this section can be found in Appendix A.1.

## 2.1 The data sets

Data sets were generated by selecting $C$ cluster centres, and generating elliptical clouds of points around these centres. The centres were selected randomly from the interval [0,1), and the standard deviation of the cluster in each dimension from [0,0.4). Each data set contained approximately 3000 points in 40 dimensions. 400 sets were generated in total: 100 for each of $C \in \{2, 3, 5, 7\}$. Figure 2.1 shows a principal component analysis of a dataset where $C = 7$, illustrating in two dimensions how the points are distributed in the clusters, and how separated those clusters are.

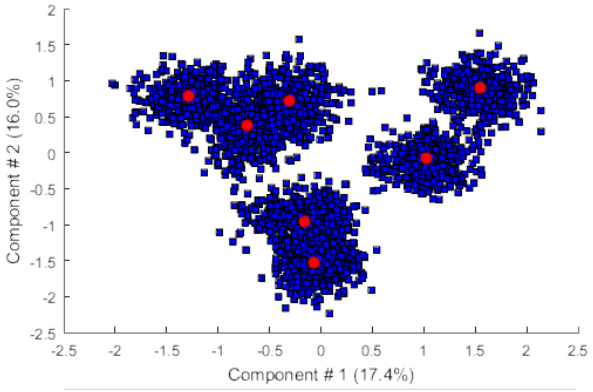


**Figure 2.1: A principal component analysis of an artificial data set used in this experiment**

Artificial data sets were created instead of using real data, because this allows more control over the experiment. Many characteristics of the data are unrealistic to expect from real data, in particular: the points are spread over roughly the same range in every dimension, and the clusters are uncorrelated. Generally, some pre-processing of data would applied before applying a clustering algorithm to a problem (Liland, 2011), but for testing purposes data sets can be designed such that this is unnecessary. Additionally, modifications of the k-means algorithm are common to deal with constraints of a problem (Wagstaff, Cardie, Rogers, Schrödl, et al., 2001). For example, when the clusters are expected to be correlated, Mahalanobis distance (Mahalanobis, 1936) can be used instead of euclidean distance to account for this. The simplifications in this experiment make the data sets ideal for the k-means algorithm, meaning that the proposed strategy can be assessed in a sterile environment free from the confounding factors found in real data.

Another advantage of artificial data sets is that they can be easily created with a specific number of cluster centres, providing a 'gold standard' answer by which to evaluate success. A comparison only to the gold standard would do nothing to assess the viability of the algorithm in contrast to the alternatives: so in this experiment there is also a comparison to the silhouette method. On the other hand, since the methods being compared incorporate different information to produce an answer, the results could be quite different, and without a gold standard it would be difficult to identify which is superior.

## 2.2 Attributes

The strategy in this thesis requires that attributes are present for each sample, outside of the feature vector proper. If the attributes were distributed uniformly, there would be no additional information with which to evaluate the clustering result. Fortunately, natural measurements of frequency distribution often follow a highly skewed distribution. For example the Zipfian distribution of words in natural language (Li, 2002), populations of cities (Auerbach, 1913), or TV viewing figures (Eriksson, Rahman, Fraile, and Sjöström, 2013); or the first digits of numbers from natural data following Benford's law (Hill, 1998). If such an unbalance is present, then cluster results can be evaluated on the basis of the likelihood that attribute allocation to clusters is random.

The characteristics of an attribute could have implications for the results. For this reason, the first part of the experiment is to compare a few attributes to assess their suitability for this method. To that end, five attributes were designed to be

tested:

1. A binary attribute, and assigned completely randomly to each point ($P(a = 1) = 0.5$). A difference between clusters is what the chi-squared test is testing for, and since there is no difference between clusters, this attribute is clearly unhelpful with the proposed. It was included in the experiment as a baseline comparison.

2. An attribute that is assigned a binary value based on a different probability for each cluster ($P(a = 1) = P_c$). The clusters' probability values $P_c$ were similar, drawn from a binomial distribution ($\mu = 0.5, \sigma = 0.05$). This attribute is designed to be only slightly informative.

3. An attribute that is assigned a binary value with probability proportional to the points' value in one dimension ($P(a = 1) = \frac{x+1.5}{4}$). Interestingly, attributes could be assigned in this way to a data set which didn't contain nominal attributes to begin with. This possibility will be discussed further later.

4. An attribute that was drawn from 5 different values, where for $n \in (0, 1, 2, 3) : P(a = n) = (1-P(a = n-1))*P_c$ else $a = 4$, using the same $P_c$ as in the second attribute ($\approx 0.5$). The same attributes are most common for all clusters, however the exact distribution for each cluster differs slightly. This style of distribution was chosen since it is similar to the aforementioned Zipfian distribution, which has been shown to be common in the distributions of natural frequencies (Rousseau, 2002; Li, 2002).

5. Finally, a binary attribute that is completely homogeneous over the points generated from each centroid ($C_n = n\%2$). It would be expected that this is an extremely informative attribute.

Combinations of these attributes are also tested in this experiment, firstly by comparing mean histograms: comprising of the mean p-values of all four informative attributes or the two most suitable attributes. Secondly, a chi-squared analysis of the three-dimensional contingency tables produced by combining the two most suitable attributes is used, analysing whether the table is independent in all three dimensions. The multivariate analysis has potential to be especially powerful, since it will also take into account interaction effects between the two attributes.

It is hoped that by combining information from multiple attributes the prediction can be improved, although it is also possible that the prediction is only as good as that of the best attribute.

## 2.3 Histogram analysis

While the result of the method described in this thesis is a histogram of p-values, a simple way to automatically interpret that histogram will make the method less subjective, and more useful in autonomous applications. An automatic interpretation of the histograms needs to *either* select $k$, or reject the histogram as not useful. Since the attribute assignment is random, there is no guarantee that attributes will be informative every time, and so it is preferable for the interpreter to reject histograms that do not contain a clear dip or are otherwise ambiguous.

There are many ways that this analysis could be done, but in this experiment a very simple peak detector is used. Unlike in the cryptography example given in Section 1.7, where adjacency between letters is unimportant; partitions for similar values of $k$ also show similar p-values. This means that the histograms will show a curved shape rather than having a single value much lower than all others. To make the shape clearer, the histogram is first smoothed with a moving average filter. Although the k-means algorithm and chi-squared analysis are repeated 50 times for each $k$ for each data set, there is still roughness in the histograms. Then, from the minimum value, up to a given diameter it is checked that the p-value does not decrease when moving to the next closest $k$, to check that the desired shape is present. If the minimum p-value is not a dip up to this radius, or is greater than 0.5, then the histogram is rejected.

Some preliminary testing was performed to select the parameters of this peak-detector. A filter width of 5 was selected, since in testing it was found that a larger value was more likely to move the minimum. It was also found that peak-diameter being much larger than the filter width resulted in many rejections of 'almost good' histograms, and so 5 is also used. Finally, ties are resolved by selecting the

lowest $k$ to have that p-value, which was found to be especially practical when many p-values were 0 (see Section 4.3 for a discussion of this case). If less than 2 means were non-zero values, then the histogram was rejected, since no real decision can be made from an empty histogram.

## 2.4 Details concerning the k-means algorithm

This experiment uses a variation of the standard algorithm called "k-means++", which has been shown to consistently converge faster, and reach a lower variance in the final clustering (Arthur and Vassilvitskii, 2007). This is achieved by carefully selecting the initial centres, choosing points from the data with probability proportional to their distance from the closest centre already selected ($P(x) = \frac{D(x)^2}{\sum_{x \in \chi} D(x)^2}$). This results in a configuration in which the centres begin more evenly distributed within the data, facilitating more reliable convergence.

Squared euclidean distance is used as the measure of similarity, since this is standard practice. No limit on the number of iterations of the algorithm is used. Finally, if they occur, empty clusters are re-initialised as a singleton cluster containing the point furthest from the centre in the last iteration. This is done to avoid having any empty clusters at convergence, which make no sense to include in the context of choosing k, as well as causing problems for the chi-squared test.

## 2.5 The validity of the chi-squared test

Other, less prevalent, statistical tests for contingency tables were initially considered. Many are only suitable for tables of certain sizes, for example Fisher's exact test (Fisher, 1922), which is more accurate than chi-squared but only applicable to 2-by-2 tables. Others are more computationally intensive, for example the extension of Fisher's test for larger tables (Mehta and Patel, 1983). There is no reason that a G-test (Sokal and Rohlf, 1981) could not be used, however it suffers from similar issues to Pearson's chi-squared test, and is far less prevalent since the test statistic was historically more difficult to calculate (McDonald, 2014).

The chi-squared test relies on an approximation that is inaccurate at with lower sample-sizes (<50), where it tends to underestimate p-values (Yung-Pin, 2011). For this reason, this trick should not be used on very small data-sets. Since p-values are being compared to each other rather than to a critical value, they do not need to be correct approximations, as long as they retain their topological characteristics when compared. That being said, there is no guarantee that this is the case, since examples can be constructed where the test over- or underestimates p-values for the same sample size (Yung-Pin, 2011).

According to Yates, Moore, and McCabe (1999), the chi-squared test is considered valid when: "No more than 20% of the expected counts are less than 5 and all individual expected counts are 1 or greater". This could be violated when testing large numbers of small clusters, or with attributes that can take many values, as these characteristics would lead to many low expected counts. A user of this method would be in dangerous territory if the count of the least common attribute value (or the total for the least common 20% of the attribute values) was less than five times the maximum $k$ tested. Outliers that create singleton clusters, or attribute values that occur very rarely would also invalidate the chi-squared test, since these would lead to expected counts of less than 1.

Due to the design of this experiment, none of the above issues are present, but users should be wary of data-sets with distant outliers, many uncommon attribute values, or small samples.

## 2.6 Comparison

For each number of centroids $C \in \{2, 3, 5, 7\}$, $k \in \{2, 3, 4, ..., 19\}$ is tested for 100 data-sets. The algorithm recommends values of $k$, and for each attribute or combination method, the mean deviation from $C$ is be used as a measure of success. The results of the most suitable attribute or combination is then compared with the results of the silhouette method on the same data sets. A Kolmogorov-Smirnov test can be used to compare the results for a statistically significant difference (Massey Jr, 1951). This is a non-parametric test for comparing histograms, which is important since there is no reason to assume that the distribution of $k$ values will follow a known distribution.

The silhouette method is a distance based strategy as discussed in Section 1.4, it is completely unambiguous in the interpretation of the result. It is popular (Berkhin, 2006), and since it doesn't require a global calculation, it is relatively unintensive computationally, and so ideal for many repeated trials.

# 3 Results

## 3.1 P-value histograms

The histograms of p-values generally fell into three categories, illustrated in Figures 3.1, 3.2, and 3.3. Figure 3.1 shows a useful histogram, that has a curved shape, with a dip at $k = 8$. It also shows the skew that was typical of these plots, where p-value increases only gradually at values of $k$ to the right of the dip.

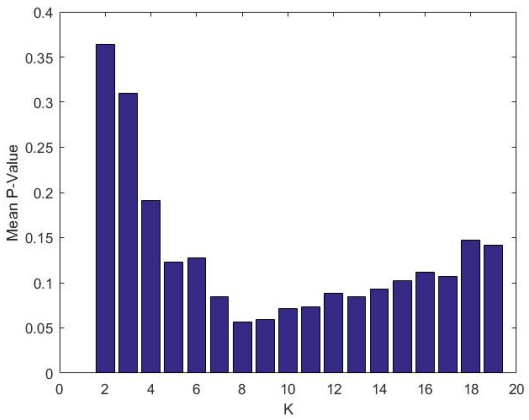


**Figure 3.1: A useful histogram of p-values**

Figure 3.2 shows an uninformative histogram that would be expected from a random attribute. With a random attribute, the p-values are generally high and there is no minimum dip. Some of the random plots showed a dip, when by chance the distributions of attributes were different in different clusters.

Figure 3.3 shows a histogram containing many average p-values of 0. These plots were an initially unforeseen issue with the proposed algorithm. A single p-value of 0 means that the distribution of attributes is incredibly unlikely to be independent of cluster in that partition. When the attributes are very unevenly distributed, this can be the case for many different $k$ values. When a plot looks like Figure 3.3, all that can be done is to take the smallest $k$ for which the p-value is 0, which is normally around $C$. The justification for this being the assumption that the shape in Figure 3.3 is an extreme case of the skew in Figure 3.1, and so the first 0 is roughly equivalent to the dip that is being searched for.

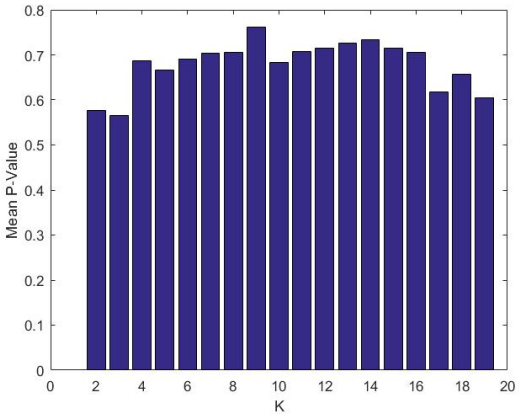


**Figure 3.2: A histogram of p-values from a random attribute**

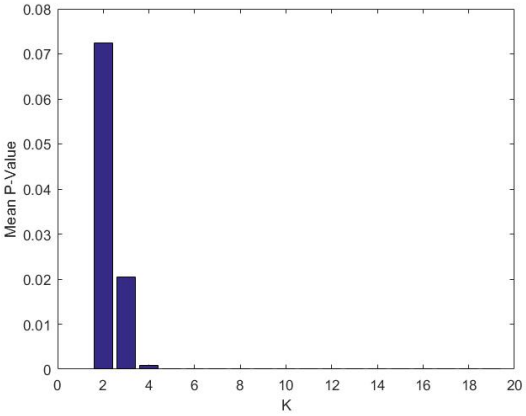


**Figure 3.3: A histogram of p-values from a very irregularly distributed attribute**

## 3.2 Attributes

Table 3.2 shows the mean p-value recommended with each attribute in each case. Table 3.1 shows the mean deviation from the number of centroids

**Table 3.1: Mean deviation from the number of centroids, and counts of histograms rejected for each attribute**

| | Number of Cetroids | | | | | Total Rejections |
|---|---|---|---|---|---|---|
| | 2 | 3 | 5 | 7 | All | |
| Random | 2.86 | 3.39 | 3.24 | 3.93 | 3.36 | 163 |
| Centroid Probability | 0.81 | 1.22 | 2.15 | 2.41 | 1.65 | 68 |
| Correlated | 3.50 | 1.83 | 1.79 | 1.94 | 2.26 | 78 |
| 5-value | 1.12 | 1.67 | 1.84 | 2.32 | 1.73 | 51 |
| Homogeneous | - | - | - | - | - | 400 |
| Multivariate of 2 | - | - | - | - | - | 400 |
| Mean of 4 | 0.73 | 1.27 | 2.18 | 2.55 | 1.68 | 105 |
| Mean of 2 best | 0 | 1.10 | 1.93 | 2.48 | 1.38 | 100 |

$C$, as well as the number of histograms that were rejected by the algorithm. The best two attributes that were selected for the combinations were the constant centroid probability attribute and the 5-value attribute. Firstly, the random attribute performed poorly since there was no information in the attribute; as expected it gave the highest deviation. Table 3.1 shows that only 163 of the 400 histograms were rejected by the peak detector. One would hope that more of these plots would be rejected if there is no information to be garnered from the attribute, however many of these plots did show a curved shape. It can also be seen in 3.2 that the mean p-value increased slightly with number of centroids.

**Table 3.2: Mean $k$ recommended for each attribute**

| | Number of Cetroids | | | |
|---|---|---|---|---|
| | 2 | 3 | 5 | 7 |
| Random | 4.86 | 5.76 | 6.11 | 7.43 |
| Centroid Probability | 2.81 | 4.15 | 6.83 | 9.09 |
| Correlated | 5.50 | 4.80 | 6.50 | 8.08 |
| 5-value | 3.11 | 4.57 | 6.73 | 8.91 |
| Homogeneous | - | - | - | - |
| Multivariate | - | - | - | - |
| Mean of 4 | 2.73 | 4.26 | 7.05 | 9.42 |
| Mean of 2 best | 2 | 4.05 | 6.83 | 9.25 |

The homogeneous attribute and the multivariate analysis suffered from the same problem: the p-value plots showed only 0s. For the homogeneous attribute, every single data-set with almost every single $k$ produced a partition of the attributes that was determined by the chi-squared test to be completely unlikely to be random. In essence, the attribute was *too informative* for this method. The chi-squared test in three dimensions was testing for independence in every dimension. Since both attributes individually were not independent of the clusters, (and by extension not independent of each other), these three dimensional contingency tables were far more informative than the two dimensional tables. This meant that the p-values were given as 0.

The correlated attribute was the worst performing of the remaining attributes, especially in the $C = 2$ case, where the mean prediction was 5.5. This attribute encoded the distance information of a single dimension, and it is not surprising that this was not sufficient to determine $k$ within a 40 dimensional dataset. Rather, this attribute would prefer the partitions where the clusters are furthest apart in the selected dimension, and therefore the $k$ where this is more likely to happen. This value may be tenuously related to $C$, but this experiment shows that the relation is not strong enough to provide a good performance.

The two attributes with constant and similar distributions over each cluster performed better than the other attributes. These attributes showed the lowest deviation from $C$, and also the lowest numbers of rejected histograms.

The combination of the four informative attributes by averaging histograms was essentially a combination of 3 attributes, since the homogeneous

attribute produced a histogram of 0s. It performed about as well as the two best individual attributes. The mean of the two best attributes appears to be the best way to use the available attributes to choose $k$. It seems that averaging histograms of attributes is a good way to seek consensus and improve the estimation of $k$. It also allows a result to be shown when *either* attribute is ideal but the other is too informative.

Overall, even in the best attributes, there was a clear tendency to overestimate $k$, which seemed to be more prevalent at higher numbers of centroids. The peak finder seemed to be effective for rejecting attributes that were too dependent on cluster, but was less effective at rejecting random histograms.

## 3.3 Comparison to the silhouette method

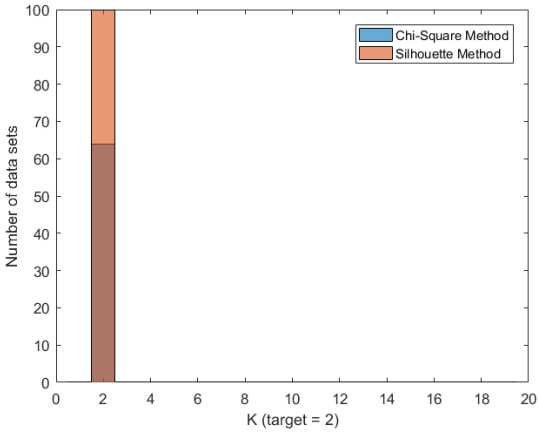


**Figure 3.4: $k$ values recommended by both strategies when $C$=2**

**Table 3.3: Mean $k$ and deviation from $C$ for the silhouette method on the test data.**

| | Number of centroids | | | |
|---|---|---|---|---|
| | 2 | 3 | 5 | 7 |
| Mean $k$ | 2.00 | 3.00 | 5.18 | 7.12 |
| Deviation from $C$ | 0.00 | 0.14 | 0.56 | 0.72 |

Figures 3.4, 3.5, 3.6, and 3.7 show the difference in performance between the silhouette and chi-squared methods of choosing $k$ for $C = 2, 3, 5, 7$.

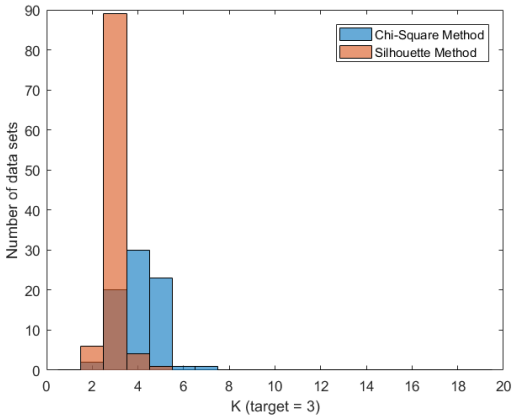


**Figure 3.5: $k$ values recommended by both strategies when $C$=3**

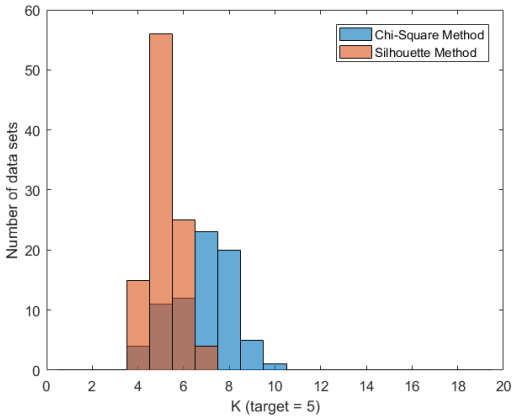


**Figure 3.6: $k$ values recommended by both strategies when $C$=5**

A Kolmogorov-Smirnov test shows that the difference between methods when $C = 3, 5, 7$ was statistically significant ($p<0.000001$). Both methods only reported $k = 2$, for $C = 2$, however the histogram peak detector failed to produce an answer in 36 of these cases. Compared to the chi-squared method's mean deviation from $C$ of 1.38, the silhouette method's was just 0.36. It is clear from the given plots, and comparison with Table 3.3 that the silhouette method had a far superior performance on this data: the $k$ values were both more tightly distributed, and less offset from the target value.

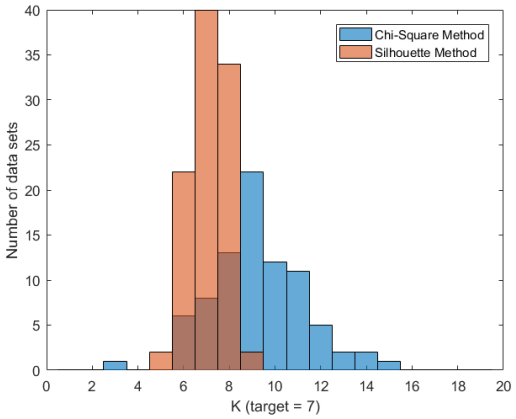


**Figure 3.7: $k$ values recommended by both strategies when $C$=7**

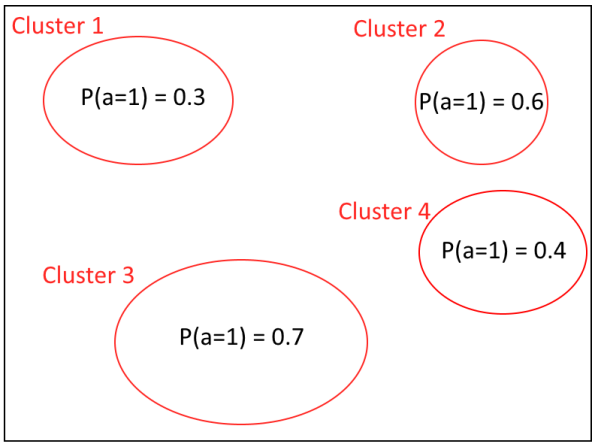


**Figure 4.1: An example of four separated clusters in a dataset**

# 4 Discussion

## 4.1 The skew

Hypothetically, if the k-means algorithm with $k = C$ produces clusters that match exactly to the centroid from which the points were generated, then the underlying relationship between cluster and attribute will be reflected in this clustering. When clustering with smaller values of $k$, the resulting partition will likely have combined some of these clusters, leading to (on average) a less distinct attribute distribution and so higher p-value from the chi-squared test. For example, in Figure 4.1, if clustered with $k = C - 1$, clusters 2 and 4 may be combined into one cluster where $P(a = 1) \approx 0.5$ (assuming the clusters are roughly the same size). This would result in a lower chi-squared statistic, since the difference in attribute counts between clusters 2 and 4 has been lost, and so this partition is would give a higher p-value.

Moving on to consider clustering with $k = C + 1$, it is likely that the partition will be very similar to that of $k = C$, with a difference such as one cluster being split into two, or two nearby clusters being divided into 3. Such partitions will still show the relationship between cluster and attribute fairly well. For example, in Figure 4.1, if Cluster 3 is split into two clusters, then there will now be two clusters where $P(a = 1) \approx 0.7$. The proof in Appendix A.2 shows that if the two sub-clusters are of equal size, then the chi-squared statistic cannot decrease, and will increase if the attributes are not distributed evenly over the two clusters. If the chi-squared statistic stayed the same then the p-value would be slightly higher due to the different number of degrees of freedom in the two tables, but an uneven split of attribute values would cause a lower p-value. The provided proof applies only to some cases, but it illustrates that the tendency for p-value to increase only a little when clustering with a higher $k$ is caused by sub-divisions of the clusters producing *almost* equally unlikely partitions of the data. This then leads to the skewed histograms seen in this experiment (3.1).

This also explains why the strategy employed in this thesis tended to overestimate $k$ when compared to both $C$ and the silhouette method. $k = C + 1$ or $C + 2$ will produce distributions of attributes approximately as unlikely to be independent as $k = C$. The implication of this explanation is that the over-estimation of $k$ would be less present in data that is less well separated, since k-means would be less likely to produce sub-clusters when $k > C$. However, the wisdom of applying the k-means algorithm to data that does not have an underlying grouped structure is questionable.

In the case that a cluster is split into two clusters for higher $k$, and the attributes are distributed unevenly over these sub-clusters, it is arguable that the attribute information is revealing additional structure in the data. The uneven distribution of the attribute could be taken as an indication that

this cluster actually *should* be split into two. This decision would have to be context dependent, and if a user wants attributes to be differentiating between clusters then they should perhaps be incorporating them into their original clustering algorithm. Otherwise, there would be no guarantee that the cluster be split in two when the k-means algorithm is repeated. There is no evidence that a difference between result of a distance based metric and this method could be a sign that the attributes are differentiating variables, since such a difference was present in this experiment where attribute probabilities were constant within clusters.

## 4.2 Peak detection

The peak detector used in this experiment was a rudimentary way of interpreting histograms. Especially its rejection of the random histograms was unsatisfactory. Histograms for which the minimum p-value was above the fairly arbitrary threshold of 0.5 were rejected, however this threshold was designed to not reject any meaningful plots, rather than to be an ideal discriminator. A simple threshold will not be ideal because the height of the minimum p-value is affected by many factors, especially $C$ and sample size. An investigation into some critical values could produce some interesting results, but effort may be better invested into a more complex automatic interpreter of the plots. A neural network approach could work, nevertheless obtaining sufficiently diverse training data to make a general classifier would be challenging, not to mention that the more complex the method becomes, the harder it would be to replicate.

While an automatic histogram analysis method would be vital if this strategy were to be applied in autonomous domains, the histogram of p-values contains more information than a single value. A lower minimum suggests a $k$ stronger relationship between cluster and attribute, while a steeper dip suggests less flexibility in $k$. The dip may even be a different shape depending on characteristics of data that were not varied in this experiment, such as the shape of, or distance between clusters.

## 4.3 Attributes

The informativeness of an attribute was more vital for its usability than expected, since it emerged that this algorithm didn't work if an attribute was *too* informative with relation to the clusters. That something can be 'too informative' is counter-intuitive, but makes sense when we consider the underlying mechanisms of the method. There is a limit to the precision with which we convert chi-squared statistics to p-values, since the value is normally compared to a threshold, so statistics software is not required to make precise conversions for very small p-values ($<0.001$). In this experiment, the minimum non-zero p-value was 0.000001, but that was not precise enough to make use of the homogeneous attribute, or multivariate combination of attributes.

As mentioned in Section 4.2, the histogram can have uses besides the extraction of $k$. If a histogram of p-values is made that is all or mostly 0, this is an indication that there is a very strong relationship between cluster and attribute. This could be taken as an indication that perhaps it is worth the effort to include this categorical variable in feature vectors, using methods described by Huang (1998). When the chi-squared tests can't be used to choose k, this will still tell the user something about the attribute that may be useful.

Correlated attributes were interesting to consider, since they essentially encode distance information into an attribute. The problem with this is that p-values depend on how distant the clusters are from each other in the dimension that was encoded. Incorporating multiple dimensions into a single binary attribute (eg. $P(a = 1) = \frac{x+y}{2}$) would only differentiate along a different line in the vector space. Using different attributes for different dimensions, or encoding dimensions into a multivariate attribute would be an unnecessarily stochastic and complex way to replicate distance based methods. The unsuitability of this attribute is problematic for this method's versatility, since it is highly likely in real-life applications that attributes will be have relationships with one or more numerical features. It appears from this experiment that nominal attributes that are directly related to hidden cluster groups are more suitable than attributes that are only related to the groups because of a relationship to the data. Assessing a partition of data using attributes that are entirely dependent on the data is theoretically problematic because there is no new information being used. Instead, the partition is being evaluated with some of the same infor-

mation that it was created with. These theoretical issues appear to have manifested in an overall poor performance of the correlated attribute in this experiment.

Producing separate histograms and then averaging them was a successful means of combining multiple attributes. Combining the two best attributes produced better predictions of $k$ than either individually. When two histograms agree, the new average histogram will show the same result. When one histogram has a clear dip but the other does not (either because it is too flat or the p-values are too low), the new histogram will reflect the decision of the better original histogram. Where two histograms disagree, the average histogram will either be a compromise between the two or unclear, either of which could conceivably be the correct course of action depending on whether the histograms disagree because of randomness or because of a problematic characteristic of attributes. The downside of combining attributes is that the characteristics in the histogram become more difficult to ascribe to the underlying relationships between attribute and data, without also looking at individual plots.

Overall, it appears that the attributes must meet specific requirements in order to provide an estimate of $k$, and as was seen in the experiment, that prediction may not be accurate.

## 4.4 Comparison with existing methods

The comparison with the silhouette method showed the chi-squared heuristic to be unquestionably inferior under these experimental conditions. The k-means algorithm aims to optimise the within cluster variance of a data-set, a parameter that is explicitly used by the silhouette method to evaluate the clustering. The chi-squared method makes use of additional information, but forgoes this distance information. This experiment has shown that the distance information can provide a better estimate of $k$ than the attribute information. This data was designed to be ideal for the k-means algorithm, but elliptical clusters with distance between them also make these ideal conditions for distance based metrics for choosing $k$. It appears that situations that lend themselves to the use of the k-means algorithm, will also be more appropriate for distance based methods for choosing $k$. It could be that attributes become more useful compared to distance in less ideal circumstances, but those are also situations where k-means is likely not the best choice of clustering algorithm, since it relies on distance measures to cluster.

In autonomous systems, combining a variety of techniques and sources of information in the calculation of a result generally allows an agent to make more robust estimates of parameters (Parker, 1995). Diversity in the methods used is an important part of this process, meaning that new techniques need not replace older techniques to be useful. Compared to other methods for estimating $k$, the chi-squared heuristic makes use of entirely different information. Where extra information is present, this strategy could see use as an orthogonal method, *alongside* distance based measures to integrate this additional information into a decision.

## 4.5 Potential applications to other problems

The method employed in this thesis is non-parametric, so it's strengths may well lie where distance-based methods are unhelpful for evaluating performance. This is the case with density-based or non-parametric algorithms, since they find non-linearly-separable clusters. The k-nearest-neighbours algorithm (Altman, 1992) for example, also has a parameter $k$, defining how many neighbours to consider when determining class membership. DBSCAN needs a parameter $\epsilon$ that defines the size of the neighbourhoods of points. With DBSCAN it is still typical to use an 'elbow' method for choosing $\epsilon$ (Ester et al., 1996; Ester et al., 1998; Schubert, Sander, Ester, Kriegel, and Xu, 2017). Additionally DBSCAN has no means by which it can incorporate nominal attributes into the clustering. These characteristics mean that there is potential for using a chi-squared heuristic for parameter selection.

An investigation into the use of chi-squared tests for parameter selection for these other algorithms could be more fruitful, depending on whether similar problems are encountered as in this paper.

### 4.6 Conclusion

An assessment of a variety of attributes made some interesting discoveries, for example finding that an attribute could be *too* informative for use with this method. Unfortunately, the method is outclassed by distance based strategies when it comes to selecting $k$ for k-means clustering, where it tends to overestimate $k$. It is hypothesised in this discussion that because of the integral role within-cluster variance plays in k-means, data that is suitable for the algorithm will necessarily also be suitable for distance based methods of assessing its performance. There is clearly some merit to the use of chi-squared heuristics for parameter selection in clustering; and further research might explore applications with different algorithms.

## References


H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

A. Arroyo, A. Herrero, and V. Tricio. Analysis of meteorological conditions in spain by means of clustering techniques. *Journal of Applied Logic*, 24:76–89, 2016.

D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

F. Auerbach. Das gesetz der bevölkerungskonzentration. *Petermanns Geographische Mitteilungen*, 59:74–76, 1913.

P. Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine learning*, 56(1-3): 9–33, 2004.

M. Eriksson, S. H. Rahman, F. Fraile, and M. Sjöström. Efficient interactive multicast over dvb-t2-utilizing dynamic sfns and parps. In *Broadband Multimedia Systems and Broadcasting (BMSB), 2013 IEEE International Symposium on*, pages 1–7. IEEE, 2013.

M. Ester, H. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

R. A. Fisher. On the interpretation of $\chi$ 2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.

R. Ganesan and A. T. Sherman. Statistical techniques for language recognition: An empirical study using real and simulated english. *Cryptologia*, 18(4):289–331, 1994.

T. P. Hill. The first digit phenomenon: A century-old observation about an unexpected pattern in many numerical tables applies to the stock market, census statistics and accounting data. *American Scientist*, 86(4):358–363, 1998.

X. Hu and L. Xu. Investigation on several model selection criteria for determining the number of cluster. *Neural Information Processing-Letters and Reviews*, 4(1):1–10, 2004.

Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

A. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

D. J. Ketchen Jr and C. L. Shook. The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal*, pages 441–458, 1996.

W. Li. Zipf's law everywhere. *Glottometrics*, 5: 14–21, 2002.

K. H. Liland. Multivariate methods in metabolomics–from pre-processing to dimension reduction and statistical analysis. *TrAC Trends in Analytical Chemistry*, 30(6):827–841, 2011.

S. P. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory, published in 1982*, 28(2):129–137, 1957.

P. C. Mahalanobis. On the generalised distance in statistics. *Proceedings National Institute of Science, India*, 2(1):49–55, 1936.

F. J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

J. H. McDonald. *Handbook of Biological Statistics*. Baltimore: Sparky House Publishing, 2014.

C. R. Mehta and N. R. Patel. A network algorithm for performing fisher's exact test in r× c contingency tables. *Journal of the American Statistical Association*, 78(382):427–434, 1983.

B. Mirkin. Choosing the number of clusters. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):252–260, 2011.

J. R. Parker. Voting methods for multiple autonomous agents. In *Intelligent Information Systems, 1995. ANZIIS-95. Proceedings of the Third Australian and New Zealand Conference on*, pages 128–133. IEEE, 1995.

D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Icml*, volume 1, pages 727–734, 2000.

K. S. Pollard and M. J. Van Der Laan. A method to identify significant clusters in gene expression data. *Proceedings, SCI (World Multiconference on Systemics, Cybernetics and Informatics)*, V. II:318–325, 2002.

G. Punj and D. W. Stewart. Cluster analysis in marketing research: Review and suggestions for application. *Journal of Marketing Research*, 20: 134–148, 1983.

R. Rousseau. George kingsley zipf: life, ideas, his law and informetrics. *Glottometrics*, 3:11–18, 2002.

P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

B. Y. Ryabko, V. S. Stognienko, and Y. I. Shokin. A new test for randomness and its application to some cryptographic problems. *Journal of statistical planning and inference*, 123(2):365–376, 2004.

J. Sander, M. Ester, H. P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2(2):169–194, 1998.

L. Schomaker. Technical report from the eu mantis project. *Project MANTIS*, 2017.

E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):19, 2017.

G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

R. R. Sokal and F. J. Rohlf. *Biometry: The Principles and Practice of Statistics in Biological Research*. New York: Freeman, 2 edition, 1981.

R. L. Thorndike. Who belongs in the family. *Psychometrika*, pages 267–276, 1953.

R. Tibshirani, T. Hastie, M. Eisen, D. Ross, D. Botstein, and P. Brown. Clustering methods for the analysis of dna microarray data. *Dept. Statist., Stanford Univ., Stanford, CA, Tech. Rep*, 1999.

R. J. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 73(2):411–423, 2001.

K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl, et al. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, pages 577–584, 2001.

M. P. Windham and A. Cutler. Information ratios for validating mixture analyses. *Journal of the American Statistical Association*, 87(420):1188–1192, 1992.

D. Yates, D. Moore, and G. McCabe. *The Practice of Statistics*. New York: Freeman, 1999.

C. Yung-Pin. Do the chi-square test and fisher's exact test agree in determining extreme for 2 by 2 tables? *The American Statistician*, 65(4):239–245, 2011.

# A Appendices

## A.1 Additional implementation details

The data was created in C, with random numbers generated with the drand48() function, which generated double-precision floating point values in the interval [0,1). To sample a number from a normal distribution, 12 random numbers were summed, then $(sum - 6) * \sigma + \mu$ was used. This means that the binomial distribution was technically bound by 6 standard deviations, however this is not a problem in practice since the chance of a binomial value exceeding this range is $1.97 \times 10^{-9}$.

The clusters in the data set were all the same size, hence for $C = 7$ the data set contained 2996 data points rather than 3000. It was not necessary for clusters to be the same size, but since this was true for $C = 2, 3, 5$, this was done for $C = 7$ for consistency.

Once the data was created, the remaining analysis was done in Matlab, since it already has functions for performing k-means clustering, a silhouette analysis, cross-tabulation, and a chi-squared test, as well as easy parallelism. Statistics software would likely also have most or all these functionalities.

All calculations were performed at double-precision. Clustering was performed with the Matlab function kmeans(), with no maximum number of iterations (the default is 100) and the parameter 'Start' set to 'plus' to use the kmeans++ starting arrangement of cluster centres. Chi-squared p-values were taken from the crosstab() function, and silhouette analysis performed with evalclusters(), both standard Matlab functions.

## A.2 Proof that a cluster split into 2 equally sized sub-clusters cannot result in a lower $\chi^2$ statistic

The total chi-squared statistic for a contingency table is the sum of every cell's contribution:

$$\chi^2 = \frac{(observed - expected)^2}{expected}$$

If a cluster with expected count $e$ in a cell is split into two sub-clusters of equal size, the two cells in the new contingency table will have expected count $\frac{e}{2}$. If the sub-clusters also have the same distribution of attributes, then each new cell will have observed count $\frac{o}{2}$. This will make the contribution of each new cell:

$$\frac{(\frac{o}{2} - \frac{e}{2})^2}{\frac{e}{2}} = 2\frac{(\frac{1}{2}(o-e))^2}{e} = 2\frac{\frac{1}{4}(o-e)^2}{e} = \frac{1}{2}\frac{(o-e)^2}{e}$$

If each new cell contributes half of what the old cell contributed, then the total $\chi^2$ statistic will be the same.

If sub-clusters do not have the same distribution of attributes, then the new observed counts will be $\frac{o}{2} - x$ and $\frac{o}{2} + x$ and so the chi-squared contribution of these cells will be:

$$\begin{aligned}
&\frac{(\frac{o}{2} + x - \frac{e}{2})^2}{\frac{e}{2}} + \frac{(\frac{o}{2} - x - \frac{e}{2})^2}{\frac{e}{2}} \\
&= \frac{(o - e + 2x)^2 + (o - e - 2x)^2}{2e} \\
&= \frac{(o^2 - 2eo + e^2 + 4x^2 - 4xo + 4xe)}{2e} \\
&+ \frac{(o^2 - 2eo + e^2 + 4x^2 + 4xo - 4xe)}{2e} \\
&= \frac{2(o-e)^2 + 8x^2}{2e} = \frac{(o-e)^2}{e} + \frac{4x^2}{e}
\end{aligned}$$

From this it can be seen that when a cluster is split into two even parts, the $\chi^2$ statistic cannot decrease, it can only stay constant (if x=0).

This proof only applies when the two sub clusters are the same size, since otherwise the observed and expected counts would both change and so $\chi^2$ statistic *could* decrease, if the larger sub-cluster has a lower proportional difference between expected and observed counts than the smaller cluster.