Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics and Operations Research

Including additional distortion measures in the convex k-means algorithm

Marnix Broek (571808)

Abstract

In this thesis, we evaluate if the inclusion of extra distortion measures in the convex k-means algorithm, as described by Modha and Spangler (2003), leads to higher-quality clustering. The proposed distortion measures are the Manhattan and Chebyshev distances for numerical variables and the Jaccard distance for categorical variables and text clustering. We use four datasets to assess the performance of the distortion measures. Three contain numerical and categorical variables, the other dataset consists of text documents. We implement the convex k-means algorithm of Modha and Spangler (2003) with the added distortion measures. First, we replicate the results presented in the original paper and then we present the differences the alteration makes. The replicated results are very close to the original results for the numerical and categorical datasets, whereas we encounter difficulties obtaining similar results for text clustering. We find that including the three proposed distance measures leads to a considerable increase in micro-p values for the datasets containing numerical and categorical variables. However, for text clustering, the Jaccard distance performs poorly, resulting in substantially lower micro-p values. Our findings support using the Manhattan, Chebyshev and Jaccard distances for clustering data with numerical and categorical variables. We do not find convincing results to warrant the same motivation for the Jaccard distance for text documents.

Supervisor:	dr. R.M. Badenbroek
Second assessor:	dr. W. Van den Heuvel
Date final version:	1st July 2024

1 Introduction

Clustering methods are crucial for research and applications in various disciplines, including data mining, data science, artificial intelligence and machine learning (Ezugwu et al., 2022). Different clustering techniques include Partitioning (K-means), Hierarchical, Density-based, Grid-based, Soft, Model-based and Ensemble clustering (Oyelade et al., 2019). K-means clustering is still considered the most popular clustering method, over 50 years after being introduced by MacQueen (1967) (Sinaga & Yang, 2020). The technique is used to partition a dataset into k disjoint clusters, where each data point is assigned to the cluster with the nearest centroid, the cluster's mean. It is favoured for its relatively simple implementation, efficiency and limited memory use (Morissette & Chartier, 2013). Naturally, there has been a lot of research on improving upon the original k-means clustering technique after its invention. Some notable alterations and extensions are Mini-Batch k-means (Sculley, 2010), Unsupervised k-means clustering (Sinaga & Yang, 2020), k-means++ (Arthur & Vassilvitskii, 2007) and the convex k-means algorithm (Modha & Spangler, 2003).

Because k-means clustering aims to minimize the sum of the distances between data points and the corresponding cluster centroid, the distance function used in the process is essential to the results. In the original k-means clustering algorithm, and most of its applications, the Euclidean distance is employed. In the alterations to the algorithm discussed by de Amorim (2016), various distance functions are used: weighted squared distance, cosine distance, matching dissimilarity distance and the Minkowski distance (which generalizes Manhattan, Euclidean and Chebyshev distances). In the same paper, they note that the use of a Euclidean distance function results in clusters with a bias towards circles or spheres, dependent on the dimensions. Therefore, the additional use of other distance functions makes k-means clustering substantially more flexible. However, as is also stated in the same paper, all distance functions result in a certain type of clustering bias. The Manhattan distance, for example, leads to clusters which are diamond-shaped, whilst the Chebyshev distance gives square-shaped clusters (in the two-dimensional case). It depends on the dataset at hand which distance function leads to the best results.

The original k-means clustering method is restricted to functioning only with numerical data (Ahmad & Dey, 2007). It stands to reason that in datasets, more often than not we also encounter categorical data. Different approaches to including categorical data in k-means clustering have been developed after its invention. For example, Dorman and Maitra (2022) use the Hamming distance instead of the Euclidean distance and replace the means with modes in the objective function of the k-means algorithm. Alternatively, Chan, Ching, Ng and Huang (2004) make use of the simple matching dissimilarity distance measure for categorical variables. Cordeiro de Amorim and Mirkin (2012) transform categorical variables into numerical ones and then apply the same distance function on both. Another interesting application field for clustering which can not be dealt with by regular k-means clustering is text documents. As is the case with data containing categorical variables, text document clustering requires different distance functions. As stated by Hornik, Feinerer, Kober and Buchta (2012), the use of the Euclidean distance results in large documents being overrepresented. Dhillon and Modha (2001) suggest the use of the cosine distance to remove the influence of document sizes, as is done by

Modha and Spangler (2003). Abuobieda, Salim, Binwahlan and Osman (2013) compare three distance measures, the Normalized Google, cosine and Jaccard distances with regards to their performance in text clustering. They conclude that the Jaccard distance performs the best. Pandit, Gupta et al. (2011) also state that the Jaccard distance works well for text document clustering.

In this thesis, we evaluate whether the use of alternative distortion measures can improve the results of the convex k-means algorithm, as proposed by Modha and Spangler (2003). The originally proposed distortion measures are the Euclidean distance for numerical variables and the cosine distance for categorical variables. We generalize the Euclidean distance to the Minkowski distance as a distortion measure for numerical variables, as described by Cordeiro de Amorim and Mirkin (2012). For categorical variables, we add the Jaccard distance, which is also done by Kongsin and Klongboonjit (2020). The main research question which we aim to answer in this thesis is therefore:

Can we attain higher micro-p values with the convex k-means algorithm when we also employ the Minkowski distance (for $p = 1$ and $p \to \infty$) for numerical variables and the Jaccard distance for categorical variables and text clustering?

As a first step, we replicate the results presented by Modha and Spangler (2003). Then, we consider multiple combinations of distortion functions, using the Minkowski distance with values of p equal to $1, 2$ or ∞ (Manhattan, Euclidean and Chebyshev distances) for numerical variables and using the cosine and Jaccard distances for categorical variables. For text clustering, we use the cosine and Jaccard distances. We use the same datasets as used by Modha and Spangler (2003). They contain data on the prevalence of heart disease, annual income, credit card applications and news articles. We want to see if we can improve upon the original algorithm by implementing more distance functions. As mentioned in the introduction, different distance functions are well-suited for different types of data. The proposed algorithm is not meant for a specific type of data and in the original paper, it is applied to varying types of datasets. Therefore, considering a wider variety of distance functions could be a valuable contribution to the original algorithm. We include the Minkowski distance because this will make the algorithm more flexible towards different data shapes. The motivation behind including the Jaccard distance is its potentially superior performance for document clustering. In general, expanding the tools used in the convex k-means algorithm is a straightforward extension to the original paper. The original algorithm is limited to the two distortion methods chosen by the authors, whereas other distortion methods also show very promising results. We are interested in exploring whether the additional distortion methods can outperform the original combination of the Euclidean and cosine distances in the convex k-means algorithm.

Our replication for the numerical and categorical datasets yields results which closely resemble the original results. On the other hand, it proves to be more complicated to replicate the results for text clustering. Our replicated results differ from the original results considerably. Regarding our extension, we find promising results in favour of the Manhattan, Chebyshev and Jaccard distances for numerical and categorical features. In 12 of the 15 clusterings performed

Table 1: Variables for the Heart Disease dataset

Numerical variables	**Categorical variables**
Age	Sex
Resting blood pressure	Chest pain type
Serum cholesterol	Fasting blood sugar above 120 mg/dl
Maximum heart rate	Resting ECG results
ST depression induced by exercise relative to rest	Exercise-induced angina
	Slope of the peak exercise ST segment
	Major vessels (0-3) coloured by fluoroscopy
	Normal, fixed defect or reversible defect

for the non-text datasets, the best results are found using a combination of distance measures which deviates from the original combination of the Euclidean and cosine distances. For text clustering, the results are not convincing. The Jaccard distance results in substantially lower micro-p values for all three clusterings performed.

This thesis proceeds as follows. In Section 2, we give an overview of the datasets used to run the algorithm. Then, in Section 3 we present the methodology used in the paper. This is a combination of both the original methodology used by Modha and Spangler (2003) and the new methodology we introduce as an extension. Section 4 contains the replication and extension results and we conclude our research in Section 5, where we also present some ideas for future research.

2 Data

We use the same data used by Modha and Spangler (2003). They use four datasets from the UCI Machine Learning Repository: Heart Disease (Janosi & Detrano, 1988), Adult (Becker & Kohavi, 1996), Statlog (Australian Credit Approval) (Quinlan, n.d.) and Twenty Newsgroups (Mitchell, 1999). From now on, we will refer to the Statlog dataset as Australian, following the notation of Modha and Spangler (2003). The datasets Heart Disease and Australian are relatively small, whereas the Adult and Twenty Newsgroups datasets are much larger.

The dataset Heart Disease contains 303 observations, $n = 297$ after removing observations with missing values. Each data point contains 13 attributes, 5 of which we consider to be numerical and 8 categorical, see Table 1. The observations are divided into two classes: those belonging to people with a heart disease and those without.

The dataset Adult contains 48842 observations, $n = 47621$ after removing observations with missing values. The dataset contains 14 attributes, 6 are numerical and the remaining 8 are categorical, see Table 2. The observations are grouped based on whether their annual income is above or below \$50,000.

The dataset Australian contains $n = 690$ observations and there are no missing values. There are 14 attributes, 6 of which are numerical and 8 categorical. Due to the confidentiality of the data, as the dataset concerns credit card applications, the attributes have no descriptions or meaningful names. The dependent variable is also not specified, but once again the data points are split into two classes.

The dataset Twenty Newsgroups contains 20,000 text documents. The documents are taken from 20 online newsgroups, with 1000 documents per newsgroup. From the dataset Twenty

Table 2: Variables for the Adult dataset

Numerical variables	**Categorical variables**
Age	Work class
Final weight	Education
Education number	Marital status
Capital gain	Occupation
Capital loss	Relationship
Hours per week	Race
	Sex
	Native country

Newsgroups, we use the following 10 newsgroups:

comp.sys.mac.hardware	comp.windows.x	misc.forsale	rec.autos
rec.sport.baseball	sci.crypt	sci.space	soc.religion.christian
talk.politics.guns	talk.politics.mideast		

So for the 10 newsgroups we use, we get a total of 10000 documents, $n = 9961$ of which are left when leaving empty documents out of consideration. The empty documents are signalized by checking the number of lines specified in the document, i.e. there might still be text in empty documents, but it is stated in the document that it does not have content. Modha and Spangler (2003) do not explicitly state what classes they use for validating the clusters, but the most straightforward choice is to use the original newsgroups as classes. This also aligns with the fact that they use a minimum of 10 clusters, which corresponds to the number of newsgroups. Thus, the classes used in this thesis are the original newsgroups. Because this dataset consists of text, the number of variables is much higher when compared to the other three datasets, which contain exclusively numerical and categorical variables. The reason for the high number of variables is that the variables are constructed from 1-, 2- and 3-word phrases, which we describe more elaborately in Section 4.2. This also means the degree of complexity is a lot higher, especially when combined with the relatively large number of instances.

3 Methodology

The methodology which we apply in this thesis is largely based on the methodology applied by Modha and Spangler (2003). We mostly use the same notation and introduce new notation where necessary. If we deviate from the original notation, we state this when using it for the first time. The part of the methodology which corresponds to that used by Modha and Spangler (2003) is not described as extensively as it is in the original paper. In the original paper, the distortion measure for numerical variables is the Euclidean distance and the distortion measure for categorical variables and text clustering is the cosine distance. We generalize the numerical distortion measure to the Minkowski distance (for $p = 1, 2$ and ∞) and add the Jaccard distance as a second categorical and text document distortion measure. We discuss the data model in Section 3.1 and present the used distortion measures in Section 3.2. The corresponding generalized centroids are given in Section 3.3. Then, the convex k-means algorithm is discussed in Section 3.4 and the method to determine the optimal feature weighting in Section 3.5. Lastly,

we introduce the metrics we use to assess the quality of the attained clustering in Section 3.6.

3.1 Data model

Consider a dataset in which each observation can be seen as a tuple of m component feature vectors. We then denote a data object as $\mathbf{x} = (\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m)$, where $\mathbf{F}_i$ is a component feature vector for $i \in \{1, \ldots, m\}$. For this component feature vector, we use that $\mathbf{F}_i \in \mathcal{F}_i$, with $\mathcal{F}_i$ the corresponding feature space. We only consider a_i such that $a_i \geq 1$, with a_i the dimensions of the corresponding feature space $\mathcal{F}_i$. Note that we use a_i, whereas Modha and Spangler (2003) denote the feature space dimensions as f_i. We consider four such feature spaces in this thesis:

- Euclidean feature space: $\mathcal{F}_i$ is $\mathbb{R}^{a_i}$. It can also be a compact submanifold of the aforementioned feature space.
- Spherical feature space: $\mathcal{F}_i$ is defined as the intersection of the a_i-dimensional unit sphere with the non-negative orthant of $\mathbb{R}^{a_i}$.
- Binary vector feature space: $\mathcal{F}_i$ is a space of binary vectors $\mathbf{y} \in \{0, 1\}^{a_i}$. Every dimension represents the absence (0) or presence (1) of a feature.
- Natural feature space: $\mathcal{F}_i$ is $\mathbb{N}^{a_i}$: each feature vector only contains non-negative, whole numbers.

3.2 Distortion measures

We define a variety of distortion measures to find the distortion between $\mathbf{x} = (\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_m)$ and $\tilde{\mathbf{x}} = (\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2, \ldots, \tilde{\mathbf{F}}_m)$. We define D_i, $i \in \{1, \ldots, m\}$ as a distortion measure between $\mathbf{F}_i$ and $\tilde{\mathbf{F}}_i$. For the definitions of the distortion measures, we define $\mathbf{F}_i = (f_1, f_2, \ldots, f_{a_i})$ and $\tilde{\mathbf{F}}_i = (\tilde{f}_1, \tilde{f}_2, \ldots, \tilde{f}_{a_i})$. We continue with the definitions of the Minkowski, cosine and Jaccard distances. Lastly, we present the weighted distortion measure as used by Modha and Spangler (2003).

Minkowski distance

The Minkowski distance is used with regard to the Euclidean feature space. It is defined as

$$D_i(\mathbf{F}_i, \tilde{\mathbf{F}}_i) = \left(\sum_{j=1}^{a_i} |f_j - \tilde{f}_j|^p \right)^{\frac{1}{p}}, p \in \mathbb{N},$$

and the special cases which we consider in this thesis are the Manhattan distance ($p = 1$), Euclidean distance ($p = 2$) and Chebyshev distance ($p \to \infty$). The Euclidean distance is the distortion measure used in the original paper. We generalize this to also include $p = 1$ and $p \to \infty$. For $p \to \infty$, this yields the following distance function:

$$D_i(\mathbf{F}_i, \tilde{\mathbf{F}}_i) = \lim_{p \to \infty} \left(\sum_{j=1}^{a_i} |f_j - \tilde{f}_j|^p \right)^{\frac{1}{p}} = \max_{1 \leq j \leq a_i} |f_j - \tilde{f}_j|.$$

Cosine distance

The cosine distance is used for the spherical feature space and is defined as

$$D_i(\mathbf{F}_i, \tilde{\mathbf{F}}_i) = 2(1 - \mathbf{F}_i^T \tilde{\mathbf{F}}_i),$$

following the definition of Modha and Spangler (2003).

Jaccard distance

The Jaccard coefficient is a metric to determine the similarity of two sets and the Jaccard distance is defined using the Jaccard coefficient (Levandowsky & Winter, 1971). To accommodate the usage of binary vectors, we rewrite the definition of the Jaccard coefficient. The definition of the Jaccard coefficient for sets is

$$J(S_i, \tilde{S}_i) = \frac{|S_i \cap \tilde{S}_i|}{|S_i \cup \tilde{S}_i|}.$$

Equivalently, we define the Jaccard coefficient for binary vectors as

$$J(\mathbf{F}_i, \tilde{\mathbf{F}}_i) = \frac{\mathbf{F}_i^T \tilde{\mathbf{F}}_i}{\mathbf{F}_i^T \mathbf{F}_i + \tilde{\mathbf{F}}_i^T \tilde{\mathbf{F}}_i - \mathbf{F}_i^T \tilde{\mathbf{F}}_i}.$$

The corresponding Jaccard distance is defined as

$$D_i(\mathbf{F}_i, \tilde{\mathbf{F}}_i) = 1 - J(\mathbf{F}_i, \tilde{\mathbf{F}}_i) = 1 - \frac{\mathbf{F}_i^T \tilde{\mathbf{F}}_i}{\mathbf{F}_i^T \mathbf{F}_i + \tilde{\mathbf{F}}_i^T \tilde{\mathbf{F}}_i - \mathbf{F}_i^T \tilde{\mathbf{F}}_i}.$$

For two zero vectors, we define the Jaccard distance as zero.

The feature vectors in the natural feature space are not binary, so we need to transform the feature vectors to binary vectors before applying the Jaccard distance. Let $\mathbf{F}_i$ denote the original natural feature vector, then this vector can be transformed into a binary feature vector as follows. Let $\mathbf{G}_i$ denote the new, binary feature vector. Then each vector element $g_j, j \in \{1, \ldots, a_i\}$ is defined as

$$g_j = \begin{cases} 1 & \text{if } f_j \geq 1, \\ 0 & \text{if } f_j = 0. \end{cases}$$

The motivation behind this transformation is that we are only interested in which features are common among different data points. The frequency of the features within the data points is not relevant for the Jaccard distance. Note that, in this thesis, this feature space is used for text documents. This means that we are only interested in the phrases text documents have in common, not the frequency of the phrases within the documents.

Weighted distortion measure

Following Modha and Spangler (2003), we define a weighted distortion measure based on m distortion measures $\{D_1, D_2, \ldots D_m\}$, defined for the component feature vectors of $\mathbf{x}$ and $\tilde{\mathbf{x}}$. The m distortion measures are selected from the distortion measures described in this section. The same distortion measure can be used for various component features. Consider non-negative

feature weights $\{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ which sum to 1, with $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$. The weighted distortion measure between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ is defined as

$$D^{\boldsymbol{\alpha}}(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{i=1}^{m} \alpha_i D_i(\mathbf{F}_i, \tilde{\mathbf{F}}_i).$$

Under the assumption of convex distortion measures, $D^{\boldsymbol{\alpha}}$ is a convex combination of convex functions, making $D^{\boldsymbol{\alpha}}$ convex, too. The feature weighting $\boldsymbol{\alpha}$ is adjustable.

3.3 Generalized centroids

We denote the generalized centroid $\mathbf{c}_u$ for given cluster π_u as

$$\mathbf{c}_u = (\mathbf{c}_{(u,1)}, \mathbf{c}_{(u,2)}, \ldots, \mathbf{c}_{(u,m)}),$$

with $\mathbf{c}_{(u,j)} \in \mathcal{F}_j, j \in \{1, \ldots, m\}$. It is the solution to the following minimization:

$$\mathbf{c}_u = \arg\min_{\tilde{\mathbf{x}} \in \mathcal{F}} \sum_{\mathbf{x} \in \pi_u} D^{\boldsymbol{\alpha}}(\mathbf{x}, \tilde{\mathbf{x}}). \tag{1}$$

Due to the property of $D^{\boldsymbol{\alpha}}$ being component-wise convex, we can split (1) into m separate convex minimization problems:

$$\mathbf{c}_{(u,j)} = \arg\min_{\tilde{\mathbf{F}}_j \in \mathcal{F}_j} \sum_{\mathbf{x} \in \pi_u} D_j(\mathbf{F}_j, \tilde{\mathbf{F}}_j), \quad j \in \{1, \ldots, m\}. \tag{2}$$

We present the closed-form centroid formulas for the Manhattan, Euclidean, Chebyshev, cosine and Jaccard distances. For the Jaccard distance, the derivation of the closed-form centroid formula is also given.

Manhattan distance

From Leisch (2006):

$$\mathbf{c}_{(u,j)} = \text{median}(\{\mathbf{F}_j \mid \mathbf{x} \in \pi_u\}),$$

where the median of the j-th component, $\mathbf{c}_{(u,j)}$, is found by ordering the j-th component feature vector elements in dimensions $\{1, \ldots, a_j\}$ of all data points $\mathbf{x} \in \pi_u$ and selecting the middle value for every dimension. Using $\mathbf{F}_j = (f_1, f_2, \ldots, f_{a_j})$, we can formally denote the i-th element of $\mathbf{c}_{(u,j)}$ as

$$\mathbf{c}^i_{(u,j)} = \text{median}(\{f_i \mid \mathbf{x} \in \pi_u\}), \quad i \in \{1, \ldots, a_j\}.$$

Euclidean distance

From Modha and Spangler (2003):

$$\mathbf{c}_{(u,j)} = \frac{\sum_{\mathbf{x} \in \pi_u} \mathbf{F}_j}{|\pi_u|}.$$

Chebyshev distance

From Cordeiro de Amorim and Mirkin (2012):

$$\mathbf{c}_{(u,j)} = \frac{\min_{\mathbf{x}\in\pi_u} \mathbf{F}_j + \max_{\mathbf{x}\in\pi_u} \mathbf{F}_j}{2},$$

which is the midrange of the data points $\mathbf{x} \in \pi_u$.

Cosine distance

From Modha and Spangler (2003):

$$\mathbf{c}_{(u,j)} = \frac{\sum_{\mathbf{x}\in\pi_u} \mathbf{F}_j}{||\sum_{\mathbf{x}\in\pi_u} \mathbf{F}_j||}.$$

Jaccard distance

The centroid under the Jaccard distance is computed as

$$\mathbf{c}_{(u,j)} = \text{mode}(\{\mathbf{F}_j \mid \mathbf{x} \in \pi_u\}), \tag{3}$$

where the mode of the j-th component, $\mathbf{c}_{(u,j)}$, is determined by identifying the most frequent values of the j-th component feature vector elements in dimensions $\{1, \ldots, a_j\}$ across all data points $\mathbf{x} \in \pi_u$ and, for every dimension, selecting the value appearing most frequently. Formally, again using $\mathbf{F}_j = (f_1, f_2, \ldots, f_{a_j})$, we can denote the i-th element of $\mathbf{c}_{(u,j)}$ as

$$\mathbf{c}^i_{(u,j)} = \text{mode}(\{f_i \mid \mathbf{x} \in \pi_u\}), \quad i \in \{1, \ldots, a_j\}.$$

We proceed with the proof for equation (3). See Section 3.2 for the definition of the Jaccard distance between two binary vectors.

Consider a cluster π_u. We want to prove that the centroid $\mathbf{c}_{(u,j)}$ using the Jaccard distance is given by equation (3). This is equivalent to showing that the mode minimizes the sum of the Jaccard distances from the centroid to all data points, see equation (2). Our notation for this proof differs from the general notation used in this thesis. We define binary vectors $\mathbf{V}_i = (v_{i1}, \ldots, v_{ia_j})$ for $i \in \{1, \ldots, n\}$, where $v_{ij} \in \{0, 1\}, \forall i, j$. Denote the mode vector $\mathbf{M} = (m_1, \ldots, m_{a_j})$, where each m_i is the most frequent value in dimension i across all vectors $\mathbf{V}_i$.

The sum of Jaccard distance from a set of vectors $\{\mathbf{V}_1, \ldots, \mathbf{V}_n\}$ to the centroid is given by

$$\sum_{i=1}^{n} D_j(\mathbf{V}_i, \mathbf{c}_{(u,j)}) = \sum_{i=1}^{n} (1 - \frac{\mathbf{V}_i^T \mathbf{c}_{(u,j)}}{\mathbf{V}_i^T \mathbf{V}_i + \mathbf{c}_{(u,j)}^T \mathbf{c}_{(u,j)} - \mathbf{V}_i^T \mathbf{c}_{(u,j)}}),$$

which is minimized through the maximization of

$$\sum_{i=1}^{n} \frac{\mathbf{V}_i^T \mathbf{c}_{(u,j)}}{\mathbf{V}_i^T \mathbf{V}_i + \mathbf{c}_{(u,j)}^T \mathbf{c}_{(u,j)} - \mathbf{V}_i^T \mathbf{c}_{(u,j)}} = \sum_{i=1}^{n} \frac{\sum_{k=1}^{a_j} v_{ik} c_k}{\sum_{k=1}^{a_j} (v_{ik} + c_k - v_{ik} c_k)} = T, \tag{4}$$

where c_k is the k-th element of centroid $\mathbf{c}_{(u,j)}$ (for the remainder of this proof, we denote the

centroid as $\mathbf{c}$). We first consider the case where we set all $c_k = 0$. For all k for which $m_k = 1$, we show we can achieve an improvement for our maximization by picking $c_k = 1$. We evaluate the change in the value of T (see equation 4):

$$\Delta T = \sum_{i:v_{ij}=1} \frac{1}{|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c}} - \sum_{i:v_{ij}=0} \frac{\mathbf{V}_i^T\mathbf{c}}{(|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c} + 1)(|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c})},$$

and, as we are maximizing, we want $\Delta T \geq 0$, which is equivalent to

$$\sum_{i:v_{ij}=1} \frac{1}{|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c}} \geq \sum_{i:v_{ij}=0} \frac{\mathbf{V}_i^T\mathbf{c}}{(|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c} + 1)(|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c})}. \tag{5}$$

We now use $\mathbf{V}_i^T\mathbf{c} \leq |\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c}$, which can be seen if we interpret the first term as the intersection and the second as the union of two sets, which is how the Jaccard distance is defined. Thus, we can give the following upper bound to the right-hand side of (5):

$$\sum_{i:v_{ij}=0} \frac{|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c}}{(|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c} + 1)(|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c})} = \sum_{i:v_{ij}=0} \frac{1}{|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c} + 1}.$$

Clearly, it holds that

$$\frac{1}{|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c}} \geq \frac{1}{|\mathbf{V}_i| + |\mathbf{c}| - \mathbf{V}_i^T\mathbf{c} + 1},$$

and because, by definition of the mode, we know that the set $\{i : v_{ij} = 1\}$ is larger than or equal to its complement $\{i : v_{ij} = 0\}$, we see that (5) holds. Thus, setting the value of all elements of the centroid to 1 for the dimensions in which the mode is 1 never decreases T. Through a similar derivation, we find that setting the value of the centroid elements to 1 for dimensions with mode 0 yields a decrease in T. As we are maximizing T and no further increase is possible, we conclude that choosing the elementwise mode (see equation 3) as the centroid under the Jaccard distance is optimal.

3.4 The convex k-means algorithm

Our objective is to distribute the dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ such that we attain k disjoint clusters $\{\pi_1, \pi_2, \ldots, \pi_k\}$. We assess the quality of a partitioning $\pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$ with the sum of the individual cluster distortions. We denote this by

$$q^{\boldsymbol{\alpha}}(\pi) = \sum_{u=1}^{k} \sum_{\mathbf{x} \in \pi_u} D^{\boldsymbol{\alpha}}(\mathbf{x}, \mathbf{c}_u). \tag{6}$$

The objective of the algorithm is to find the set of k disjoint clusters $\{\pi_1^\dagger, \pi_2^\dagger, \ldots, \pi_k^\dagger\}$ which minimizes (6) for a fixed $\boldsymbol{\alpha}$. We use the convex k-means algorithm as introduced by Modha and Spangler (2003) to find the clusters, see Algorithm 1. Denote the set of clusters at the end of iteration t by $\pi^t = \{\pi_1, \pi_2, \ldots, \pi_k\}$ and associated centroids by $\mathbf{c}^t$. We denote the minimum increase percentage to continue the algorithm by β. In this thesis, we use the value $\beta = 0.0001$ (a 0.01% increase). The heuristic we use to determine our initial clustering is the same as is

used by Modha and Spangler (2003), based on Dhillon and Modha (2001). This heuristic uses the generalized centroid $\bar{\mathbf{c}}$ for the entire dataset and adds normally distributed random noise to the generalized centroid for every cluster to generate the corresponding initial centroid. After initializing the centroids, every data point is assigned to the cluster with the nearest centroid. The random noise we add is drawn from a multivariate normally distributed random variable with mean vector $\mathbf{0}$ and a covariance matrix in which only the diagonals are non-zero. The diagonals are based on $0.25 * \text{abs}(\bar{\mathbf{c}})$, in which 0.25 is the degree of deviation we set for our implementation and $\text{abs}(\bar{\mathbf{c}})$ gives the vector with the absolute values of the elements of $\bar{\mathbf{c}}$. This means that the i-th diagonal is assigned the value of the i-th element of $0.25 * \text{abs}(\bar{\mathbf{c}})$.

Algorithm 1 The convex k-means algorithm

$t \leftarrow 0$ ▷ We use t to keep track of number of iterations
Initialize: π^0, $\mathbf{c}^0$ ▷ Start with a random initial clustering and associated centroids
while True **do**
    For each data point $\mathbf{x}_i, \quad i \in \{1, \ldots, n\}$, determine which generalized centroid has the smallest distance to $\mathbf{x}_i$. Find the new clustering using the generalized centroids from the previous iteration $\mathbf{c}^t$ for $u \in \{1, \ldots, k\}$:

$$\pi_u^{t+1} = \{\mathbf{x} \in \{\mathbf{x}_i\}_{i=1}^n : D^{\boldsymbol{\alpha}}(\mathbf{x}, \mathbf{c}_u^t) \leq D^{\boldsymbol{\alpha}}(\mathbf{x}, \mathbf{c}_v^t), \quad 1 \leq v \leq k\}$$

    Update generalized centroids to $\mathbf{c}^{t+1}$ using the new clusters π^{t+1}
    if $\frac{q^{\boldsymbol{\alpha}}(\pi^t) - q^{\boldsymbol{\alpha}}(\pi^{t+1})}{q^{\boldsymbol{\alpha}}(\pi^t)} < \beta$ or $t + 1 > t^{\max}$ **then**
        $\pi^\dagger \leftarrow \pi^{t+1}$
        $\mathbf{c}^\dagger \leftarrow \mathbf{c}^{t+1}$
        return $(\pi^\dagger, \mathbf{c}^\dagger)$
    else
        $t \leftarrow t + 1$
    end if
end while

3.5 Determining the optimal feature weighting

We consider a number of clusters $k \geq 2$ and use a fixed initial clustering. We denote the generalized centroid for all data points by $\bar{\mathbf{c}} = (\bar{\mathbf{c}}_1, \bar{\mathbf{c}}_2, \ldots, \bar{\mathbf{c}}_m)$, with

$$\bar{\mathbf{c}}_l = \underset{\tilde{\mathbf{c}} \in \mathcal{F}_l}{\arg\min} \sum_{i=1}^{n} D_l(\mathbf{F}_{(i,l)}, \tilde{\mathbf{c}}), \quad l \in \{1, \ldots, m\}.$$

We denote the set of possible feature weightings by

$$\Delta = \left\{ \boldsymbol{\alpha} : \sum_{l=1}^{m} \alpha_l = 1, \alpha_l \geq 0, \quad l \in \{1, \ldots, m\} \right\}.$$

Furthermore, we use the following notation for the centroid of a cluster u:

$$\mathbf{c}_u^\dagger(\boldsymbol{\alpha}) = (\mathbf{c}_{(u,1)}^\dagger(\boldsymbol{\alpha}), \mathbf{c}_{(u,2)}^\dagger(\boldsymbol{\alpha}), \ldots, \mathbf{c}_{(u,m)}^\dagger(\boldsymbol{\alpha})), \quad u \in \{1, \ldots, k\}.$$

To allow for the comparison of the amount of distortion achieved by the algorithm using different

feature weightings, we use the metric

$$\mathcal{Q}_l(\boldsymbol{\alpha}) = \left(\frac{\Gamma_l(\boldsymbol{\alpha})}{\Lambda_l(\boldsymbol{\alpha})}\right)^{n_l/n}, \quad l \in \{1, \ldots, m\},$$

with

$$\Gamma_l(\boldsymbol{\alpha}) = \sum_{u=1}^{k} \sum_{\mathbf{x} \in \pi_u^\dagger(\boldsymbol{\alpha})} D_l(\mathbf{F}_l, \mathbf{c}_{(u,l)}^\dagger(\boldsymbol{\alpha})), \quad l \in \{1, \ldots, m\}$$

and

$$\Lambda_l(\boldsymbol{\alpha}) = \sum_{i=1}^{n} D_l(\mathbf{F}_{(i,l)}, \bar{\mathbf{c}}_l) - \Gamma_l(\boldsymbol{\alpha}), \quad l \in \{1, \ldots, m\},$$

representing the average within-cluster distortion and the average between-cluster distortion for the l-th component feature vector, respectively. n_l is the number of data points with a non-zero feature vector for the l-th component. We also define

$$\mathcal{Q}(\boldsymbol{\alpha}) = \mathcal{Q}_1(\boldsymbol{\alpha}) \times \mathcal{Q}_2(\boldsymbol{\alpha}) \times \cdots \times \mathcal{Q}_m(\boldsymbol{\alpha}).$$

We minimize this function with regards to $\boldsymbol{\alpha}$ and the resulting optimal feature weighting is denoted by $\boldsymbol{\alpha}^\dagger$.

3.6 Assessing the quality of the clustering

To allow for a meaningful comparison of the clustering results achieved by the usage of different distance functions, we use the same metrics as Modha and Spangler (2003): macro-precision (macro-p), macro-recall (macro-r), micro-precision (micro-p) and micro-recall (micro$-r$). These metrics are used for comparing clusterings for a fixed k. The metrics use precision and recall, representing the degree to which the achieved clustering matches the actual classification of the data. We assign all points from a cluster to the class with which the cluster shares the most data points.

Consider a ground truth classification with c classes $\{w_1, \ldots, w_c\}$. We use the following notation:

- $a_t, \quad t \in \{1, \ldots, c\}$: The number of data points correctly classified into class w_t.
- $b_t, \quad t \in \{1, \ldots, c\}$: The number of data points incorrectly classified into class w_t.
- $c_t, \quad t \in \{1, \ldots, c\}$: The number of data points incorrectly excluded from class w_t.

Then, we use the following definitions for precision and recall, respectively, for $t \in \{1, \ldots, c\}$:

$$p_t = \frac{a_t}{a_t + b_t} \quad \text{and} \quad r_t = \frac{a_t}{a_t + c_t}.$$

We use these definitions for macro-p, macro-r and micro-p (which is equivalent to micro-r in our case, see Modha and Spangler (2003)):

$$\text{macro-}p = \frac{1}{c}\sum_{t=1}^{c} p_t, \quad \text{macro-}r = \frac{1}{c}\sum_{t=1}^{c} r_t \quad \text{and} \quad \text{micro-}p = \frac{1}{n}\sum_{t=1}^{c} a_t.$$

Table 3: Original and replicated results for the dataset Heart Disease for a fixed feature weighting

		Original results		Replicated results	
k	$\tilde{\alpha}$	$\mathcal{Q}(\tilde{\alpha})$	micro-p	$\mathcal{Q}(\tilde{\alpha})$	micro-p
2	(.5, .5)	34.64	.741	43.85	.720
4	(.5, .5)	11.21	.733	11.52	.722
6	(.5, .5)	7.69	.711	7.37	.712
8	(.5, .5)	5.20	.711	5.09	.721
16	(.5, .5)	2.38	.767	2.08	.744

4 Results

In this section, we present the results attained with the convex k-means algorithm as described in Section 3. Because we extend upon Modha and Spangler (2003), we first replicate their most important results. The initially chosen clustering is a very relevant factor for the clustering results. As noted in Section 3.4, we make use of a Gaussian initial clustering method. This means that the initial partitioning is partially random. As Modha and Spangler (2003) do not specify their exact settings for the creation of this partitioning, we attempt to reduce the randomness by using multiple seeds for the Normal distribution. We then use the average values of $\mathcal{Q}(\alpha^{\dagger})$, micro-$p$, macro-$p$ and macro-r across all seeds per feature weighting. For the algorithm using a fixed feature weighting for numerical and categorical data, we use 25 seeds. For the optimal feature weighting for numerical and categorical data and for the fixed feature weighting for text clustering, we use 5 seeds due to the increased complexity. Lastly, for optimal feature weighting for text clustering, we use 2 seeds, because the clustering process is even more complex. The fact that this part of the algorithm is a random process indicates that 100% replicability might not be achievable, especially because Modha and Spangler (2003) do not provide details on their implementation of the initial clustering method. In this section, note that we employ the term optimal to refer to the best result found within the specified computational and heuristic limitations, both when discussing optimal feature weightings and the optimal combination of distance functions. This matches the terminology used by Modha and Spangler (2003).

The remainder of this section is split into two subsections. In Section 4.1, we present the results for the first three datasets, which contain numerical and categorical variables. In Section 4.2, we give the results found for the Twenty Newsgroups dataset, containing text documents. We present our results for the replicated part from Modha and Spangler (2003) and give the results we find using our extended methodology.

4.1 Numerical and categorical data

We first present the results for the following three datasets: Heart Disease, Adult and Australian. The set of feature weightings employed for the results is equivalent to the one used by Modha and Spangler (2003):

$$\Delta = \{(\alpha_1, \alpha_2) : \alpha_1 + \alpha_2 = 1, \alpha_1, \alpha_2 \geq 0\},$$

with α_1 the weight for the numerical feature space and α_2 the weight for the categorical feature space. We use 101 feature weightings from this set as input, i.e. the weightings we use are $\{(0.00, 1.00), (0.01, 0.99), \ldots, (1.00, 0.00)\}$.. As this thesis aims to both replicate and extend upon the original results, we split the results accordingly. In Section 4.1.1, we give the replication

Table 4: Original and replicated results for the datasets Heart Disease, Adult and Australian in determining the optimal feature weighting

	Original results			Replicated results				
k	$\alpha^\dagger$	$\mathcal{Q}(\alpha^\dagger)$	micro-p	$\alpha^\dagger$	$\mathcal{Q}(\alpha^\dagger)$	micro-p	micro-p^*	micro-p^-
Heart Disease								
2	(.09, .91)	20.49	.804	(.09, .91)	20.68	.801	.831	.706
4	(.08, .92)	6.35	.815	(.09, .91)	6.55	.797	.813	.715
6	(.08, .92)	3.77	.803	(.09, .91)	3.79	.788	.800	.696
8	(.10, .90)	2.77	.800	(.11, .89)	2.65	.795	.809	.702
16	(.12, .88)	1.15	.793	(.15, .85)	1.24	.788	.815	.716
Adult								
2	(.14, .86)	68.69	.759	(.15, .85)	68.69	.759	.759	.759
4	(.10, .90)	9.90	.761	(.12, .88)	14.59	.799	.821	.761
6	(.09, .91)	5.08	.812	(.13, .87)	6.45	.786	.814	.766
8	(.11, .89)	2.75	.820	(.14, .86)	4.21	.805	.810	.762
16	(.09, .91)	1.17	.819	(.15, .85)	1.31	.808	.814	.783
Australian								
2	(.09, .91)	38.68	.829	(.10, .90)	38.90	.821	.821	.643
4	(.09, .91)	10.31	.762	(.09, .91)	9.82	.789	.827	.648
6	(.08, .92)	5.63	.832	(.10, .90)	5.50	.804	.821	.659
8	(.10, .90)	3.89	.836	(.16, .84)	3.46	.756	.823	.665
16	(.08, .92)	1.17	.829	(.12, .88)	1.25	.797	.837	.710

results and in Section 4.1.2, we give the results of our extension.

4.1.1 Replication results

We replicate the most important results from Modha and Spangler (2003). For the dataset Heart Disease, we present the results for a fixed feature weighting and those found when determining the optimal feature weighting. For the other two datasets, we only present the results for the optimal feature weighting, as that is the main focus of Modha and Spangler (2003). We also use the results for the optimal feature weighting in the later comparison across different distance functions. The replication results for a fixed feature weighting for the other two datasets can be found in Appendix B.

Table 3 gives the replicated results for the dataset Heart Disease for a fixed feature weighting and Table 4 gives the results for the optimal feature weighting for the three datasets. The variables micro-p^* and micro-p^- represent the best and worst achieved micro-p, respectively, across all feature weightings (taking the average over all seeds). We see no extreme differences when comparing our results to those achieved by Modha and Spangler (2003), so our algorithm seems to perform very similarly to the original algorithm. The same conclusion holds for the fixed feature weighting for the datasets Adult and Australian, as shown in Appendix B. For both the fixed and optimal feature weighting clusterings, our values of $\mathcal{Q}(\alpha)$ are close to the original values but not consistently higher or lower. The same holds for the micro-p values for the fixed feature weighting clusterings. However, for the optimal feature weighting, our micro-p values are lower than the originally found values in 12 of the 15 cases. Especially for the Australian dataset, the micro-p values are substantially lower on average. The micro-p values are generally reasonably close to the micro-p^* and far from the micro-p^- values, as is the conclusion of Modha and Spangler (2003). Thus, we conclude that the metric $\mathcal{Q}(\alpha)$ is useful in finding a good feature weighting in our implementation, too, with good referring to the corresponding micro-p value.

Table 5: Extension results for the datasets Heart Disease, Adult and Australian

	Euclidean and cosine distances		Optimal combination of distance functions				
k	micro-p	macro-p	micro-p	macro-p	$\alpha^{\dagger}$	Numerical distance function	Categorical distance function
			Heart Disease				
2	.801	.801	.818	.823	(.09, .91)	Chebyshev	Cosine
4	.797	.797	.814	.823	(.18, .82)	Chebyshev	Jaccard
6	.788	.805	.813	.821	(.10, .90)	Chebyshev	Jaccard
8	.795	.805	.809	.815	(.15, .85)	Chebyshev	Jaccard
16	.788	.791	.813	.814	(.20, .80)	Chebyshev	Jaccard
			Adult				
2	.759	.380	.772	.608	(.29, .71)	Chebyshev	Cosine
4	.799	.723	.799	.723	(.12, .88)	Euclidean	Cosine
6	.786	.710	.810	.750	(.18, .82)	Manhattan	Cosine
8	.805	.736	.811	.756	(.19, .81)	Manhattan	Cosine
16	.808	.747	.812	.760	(.23, .77)	Manhattan	Cosine
			Australian				
2	.821	.827	.821	.827	(.10, .90)	Euclidean	Cosine
4	.789	.794	.806	.815	(.03, .97)	Chebyshev	Jaccard
6	.804	.810	.804	.810	(.10, .90)	Euclidean	Cosine
8	.756	.781	.808	.816	(.03, .97)	Chebyshev	Cosine
16	.797	.807	.834	.836	(.03, .97)	Chebyshev	Cosine

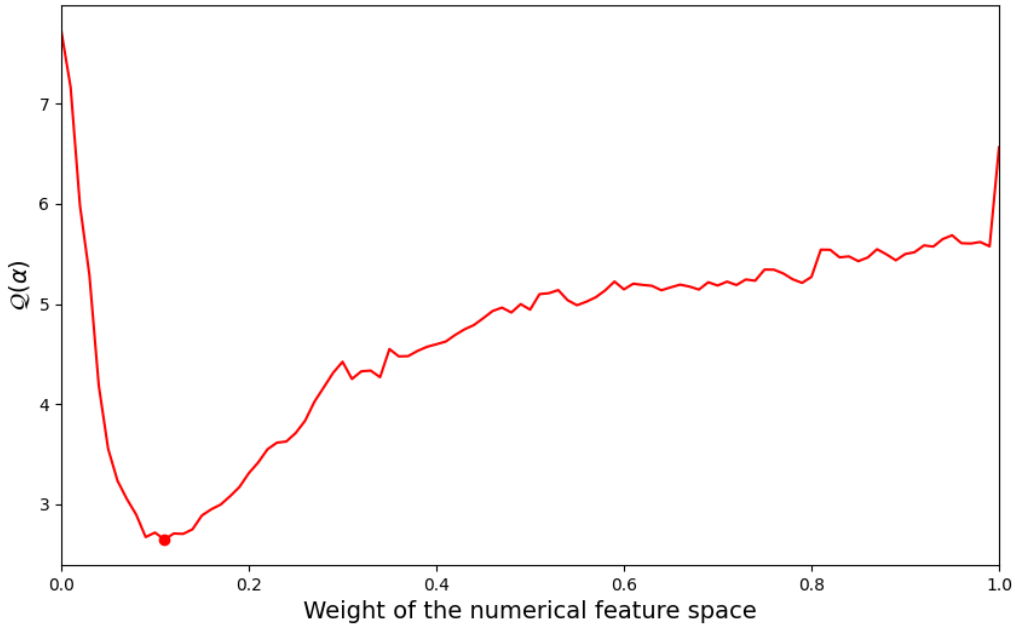
Figure 1a is our replication of the first graph in Figure 1 from Modha and Spangler (2003). It shows the value of $\mathcal{Q}(\alpha)$ for different numerical feature weightings for the Heart dataset for $k = 8$. It closely resembles the original graph and this further solidifies the similarity between our implementation of the algorithm and the original implementation.

All in all, the differences between the original and the replicated values are small enough to assume they are caused by a combination of randomness and possible small implementation differences. We do not know if Modha and Spangler (2003) use a variety of seeds or focus on one seed and we also do not have specific information on their cluster initialization technique. We expect that our implementation is, therefore, slightly different than the original. There might also be other small changes in the implementation of the algorithm. Modha and Spangler (2003) do not provide the programming code or specific details, so small differences in the outcome are not unreasonable.

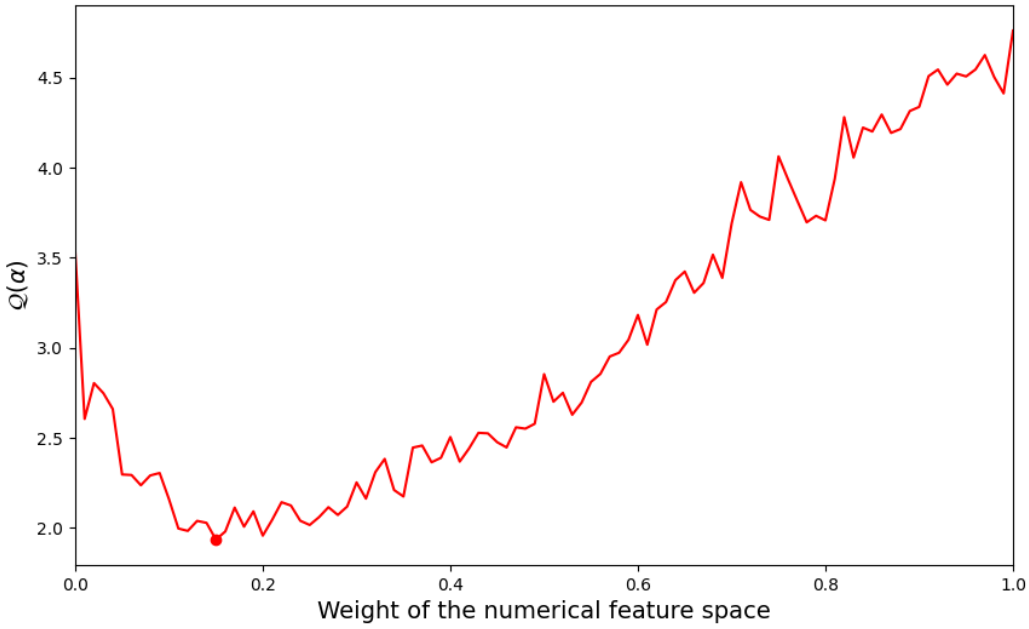
4.1.2 Extension results

We continue with the extension results for the first three datasets. In Table 5, we compare the micro-p and macro-p values found for the optimal feature weighting with the original distance functions (Euclidean and cosine distances) to the values found for the distance functions giving the highest micro-p value. We also state which distance functions give the best results and give the corresponding feature weights. The feature weights belonging to the Euclidean and cosine distances can be found in Table 4. A complete overview of the results for all combinations of distance functions is given in Appendix B. The original distance function combination is only optimal in 3 cases. This means that for the other 12 cases, we achieve an improvement in micro-p values by including the extra distance functions. We also see that the corresponding macro-p values are in all cases higher than the original macro-p values. The same observation

Figure 1: The graphs showing the values of $\mathcal{Q}(\alpha)$ for different numerical feature weightings using original and optimal distance functions, for the dataset Heart Disease and $k = 8$



(a) Products for different weights for dataset Heart Disease, $k = 8$ with original distance functions



(b) Products for different weights for dataset Heart Disease, $k = 8$ with optimal distance functions

holds for the macro-r values for all clusterings except $k = 8$ for the Adult dataset, though the macro-r values are not presented in the table. The average improvements in micro-p values for the datasets Heart Disease and Australian are larger than those found for the Adult dataset, with respective average percentual increases in micro-p values of 2.5%, 2.7% and 1.2%.

In general, all distance functions seem relevant for the algorithm when considering these datasets. The Manhattan and Euclidean distance both make part of the optimal combination 3 times, whereas the Chebyshev distance occurs in 9 optimal combinations. For the categorical distance functions, we see the cosine distance appearing 10 times and the Jaccard distance 5 times. These results are promising and strengthen our notion that including multiple distance functions can be advantageous.

An observation which stands out is the dominance of the combination of the Chebyshev and Jaccard distances for the Heart Disease clusterings. This combination is optimal for all k except 2, where this combination gives a micro-p value equal to .817, very near to the .818 found by the optimal combination of distance functions. On the other hand, the Jaccard distance is only part of one optimal combination for the Adult and Australian datasets. This indicates that different datasets abide well under the use of different distance functions, which corresponds to what is suggested in our introduction.

Another interesting insight concerns the optimal feature weightings. For the Heart Disease and Adult datasets, these tend to contain a higher weight for the numerical feature space under the optimal combination of distance functions when compared to the feature weights under the original combination, as seen in Table 4. Especially for the Adult dataset, the numerical feature weights show a substantial increase, with the exception of $k = 4$, as the original distances are optimal for this clustering. This indicates that, for these datasets, the numerical features are more informative using the additional distance functions and contribute more to the resulting clustering. For the Australian dataset, the opposite observation holds. For the clusterings with an optimal combination deviating from the original, the categorical feature weights are considerably higher. Thus, in those cases, the categorical features are more informative.

Figure 1 shows the graphs for Heart Disease with $k = 8$ under the original distance functions and the optimal combination. The minimum of the graph in Figure 1b is shifted towards the right compared to the graph in Figure 1a, which aligns with the higher numerical feature weight.

4.2 Text documents

We continue with the results for text clustering. For the text clustering, we need to make several assumptions. Modha and Spangler (2003) do not give much information on how they preprocess the text documents before performing the clustering algorithm, even though the preprocessing method influences the clustering results greatly. They refer to a list of standard stopwords from Frankes and Baeza-Yates (1992), Figure 7.5. It is not explicitly mentioned whether or not they use these stopwords or if they make use of a different list, we assume they use the list of standard stopwords and do the same. They mention that sometimes stemming is used but do not state whether they use this technique. For our implementation, we do not apply stemming. As in the original paper, we eliminate the 1-word, 2-word and 3-word phrases which are present in less than 0.64%, 0.32% and 0.16% of the documents, respectively.

Table 6: Statistics for 1-, 2- and 3-word phrases in the Twenty Newsgroups dataset

	Original statistics		Replicated statistics (method 1)		Replicated statistics (method 2)	
i	f_i	n_i	f_i	n_i	f_i	n_i
1	2583	9961	4703	9961	3563	9944
2	2144	8639	3494	9961	695	7836
3	2268	4664	6264	9961	746	4667

When we preprocess the text documents following these steps, we find the statistics as given in Table 6 under replicated statistics (method 1). We use the following notation: i represents the length of the phrases, f_i is the number of phrases of length i we keep after preprocessing and n_i is the number of documents containing at least one phrase of length i which is kept after preprocessing. It is evident that using this method, we are left with considerably more phrases and a dataset which is not sparse at all. Not only does this make clustering the data more complex, but it also gives us very different results than Modha and Spangler (2003) present. Therefore, we explore whether we should expand our preprocessing procedure using other steps which are commonly used in text clustering.

We investigate the phrases and the text documents. By looking at the structure of the news articles in the dataset, we find that there is generally a lot of information about the news article in the first lines of the documents. For example, many documents contain a line stating to which newsgroup it belongs, which is the class we want to assign it to after clustering. Most documents also contain a line stating how many lines of text it contains. However, our objective is to cluster the documents based on their text content and not on the properties given in the first lines of the documents. We therefore decide to only use the content of the news articles. By looking at the text documents, we see that the content starts after the last occurrence of "writes:", or in its absence, after "Lines:". All 9961 news articles contain at least one of the two phrases. We trim each news article accordingly, prioritizing "writes:" and only in its absence using "Lines:" as a starting point. Schubert, Lang and Feher (2021) also use the Twenty Newsgroups dataset without headers, as do Pedregosa et al. (2011).

Upon investigation of the phrases, we find many phrases which are not very informative for clustering. For example, many phrases start or end with a bracket, or contain other interpunctions or special characters which are not related to the phrase at hand. This way, one word can be divided into multiple phrases if it is combined with different special characters across the texts. This is disadvantageous for the clustering procedure, as we want to treat the same word as one phrase to accurately group the texts. The same can be said about capitalized and non-capitalized words. We also see many digits occurring as or within phrases, which do not seem very informative. Bianchi, Terragni and Hovy (2020) use the same dataset and remove all digits and punctuation in their data preprocessing. Ahmed, Tiun, Omar and Sani (2023) discuss text clustering algorithms and note that it is good practice to leave out special characters. We follow their examples and remove all digits, punctuation and special characters from the news articles. We also convert all texts to lowercase for uniformity, as is also done by Yin and Wang (2016).

These extra preprocessing steps result in the replicated statistics for method 2 as shown in

Table 7: Original and replicated results for the dataset Twenty Newsgroups for a fixed feature weighting

		Original results		Replicated results	
k	$\tilde{\alpha}$	$\mathcal{Q}(\tilde{\alpha})$	micro-p	$\mathcal{Q}(\tilde{\alpha})$	micro-p
10	(.33, .33, .33)	24.01	.593	126.75	.399
15	(.33, .33, .33)	16.20	.616	76.95	.439
20	(.33, .33, .33)	13.39	.602	56.78	.449

Table 6. Clearly, the resulting dataset is a lot more sparse than with method 1 and the number of distinct phrases has also gone down considerably. The number of 1-, 2- and 3-word phrases is still quite different from the numbers mentioned by Modha and Spangler (2003). This is not unexpected, as they do not provide many details on their preprocessing methods. The extra preprocessing actions we use are widely employed in existing literature and are even applied to the same dataset in other papers. Therefore, we consider the more extensive preprocessing method to be the most applicable for our research and use the resulting dataset as input for the algorithm for both the replication and the extension part. The set of feature weightings we explore is again identical to the one Modha and Spangler (2003) employ, containing 31 convex combinations of the points $\{(1,0,0),(0,1,0),(0,0,1)\}$. We now continue with the replication results in Section 4.2.1 and the extension results in Section 4.2.2.

4.2.1 Replication results

The replication results for a fixed feature weighting are shown in Table 7 and those for the optimal feature weighting can be found in the top section of Table 8. The differences in the replication results under a fixed feature weighting are considerable, especially when taking into account the similarity of our results for the other datasets. Our implementation of the algorithm is outperformed by that of Modha and Spangler (2003) on all fronts. The biggest differences are visible in the value of $\mathcal{Q}(\alpha)$. The micro-p values are also substantially lower. The results for the optimal feature weighting show the same trend. The values of the optimal feature weightings also show differences to a higher degree than what we find for the first three datasets. Interestingly, we see a more important role for the 2-word phrases in our results. This is likely related to the difference in phrase dictionaries caused by the data preprocessing.

Following our difficulties in mirroring the preprocessing steps used in the original paper, the relatively large differences in results are not an unexpected finding. Our implementation is likely fundamentally different from that of Modha and Spangler (2003), due to the left-out details in their description. Apart from the probable difference in the preprocessing method, we also do not know for certain which classes they assign the different news articles to. This clearly influences the micro-p values greatly, too. Another possible cause is the higher degree of randomness caused by the fewer number of seeds used. Due to the computational expense of clustering a dataset with this number of instances and variables, with dimensions of around 10,000 $\times$ 5,000, performing the algorithm for a large number of seeds is not feasible. Therefore, we might experience the repercussions of the randomness of the initialization process more heavily for this dataset. Our implementation yields acceptable results, with micro-p values which are in the

Table 8: Original, replicated and extension results for the dataset Twenty Newsgroups in determining the optimal feature weighting

	Original results			Replicated/new results				
k	$\alpha^\dagger$	$\mathcal{Q}(\alpha^\dagger)$	micro-p	$\alpha^\dagger$	$\mathcal{Q}(\alpha^\dagger)$	micro-p	macro-p	macro-r
Cosine distance								
10	(.50, .25, .25)	21.06	.686	(.67, .33, .00)	110.61	.416	.411	.416
15	(.75, .00, .25)	14.38	.656	(.58, .33, .08)	64.70	.521	.564	.522
20	(.75, .00, .25)	11.03	.664	(.58, .33, .08)	45.40	.549	.630	.549
Jaccard distance								
10				(.00, .08, .92)	293.82	.182	.298	.182
15				(.00, .08, .92)	247.52	.190	.444	.190
20				(.00, .50, .50)	202.68	.199	.467	.199

direction of the original values, but the original algorithm leads to better cluster results.

4.2.2 Extension results

We continue with the extension results for the Twenty Newsgroups dataset, shown in the bottom section of Table 8. Because the generalized centroid under the Jaccard distance (see Section 3.3) has a zero value for each feature vector element, we cannot initialize the clusters with the usual method. All data points would be assigned to one cluster with a zero vector for all feature spaces as centroid. Thus, we initialize the clusters differently for this dataset under the Jaccard distance. To avoid all clusters having the same centroid, we pick random data points as centroids of the initial clusters and then assign all data points to the cluster with the closest centroid. This way, we ensure the initial clusters are well-defined. The Jaccard distance metric is vastly outperformed by the cosine distance in every case, with very low micro-p values for the Jaccard distance clusterings. The macro-p values of the Jaccard distance clusterings for $k = 15, 20$ are the only metrics with decent values and are close to the values achieved with the cosine distance. Upon inspection of the optimal feature weightings found under the Jaccard distance, we see single words are excluded in the clustering process. This is interesting given the fact that the first feature vector is assigned the highest weight for all three clusterings under the cosine distance.

The main reasons for the poor performance of the Jaccard distance clusterings seem to be the alternative cluster initialization process and the sparsity of the dataset. As mentioned, we initialize the clusters completely at random. Simply assigning random data points as the initial centroids does not seem to be the optimal setup for good clustering results. This stands to reason, as this initialization method gives no guarantee that the initial clusters will be sufficiently dispersed. Furthermore, the sparsity of the dataset causes difficulties for the algorithm when it attempts to construct new clusters. Because there are many data points with zero feature vectors for the 2- and 3-word phrases (see Table 6), finding a new non-zero centroid proves to be complicated. We can see this in the number of iterations the algorithm takes until completion, too. Under the Jaccard distance, the algorithm usually finishes within three iterations and always within four. Contrastingly, we see upward of 20 iterations using the cosine distance. This indicates that the Jaccard distance can barely construct new clusters which lower the objective value and, thus, the initial clustering is of utmost importance for the results.

5 Conclusion

The choice of a distance function can have a big influence on the results of a k-means clustering algorithm. In this thesis, we evaluated the effect of expanding the existing convex k-means algorithm as introduced by Modha and Spangler (2003), through the addition of three distance functions: the Manhattan, Chebyshev and Jaccard distance. In particular, we were interested in seeing if the additional distance functions would lead to higher micro-p values. Because we extend upon the algorithm of Modha and Spangler (2003), we first replicated their results before examining the effect of our extension. The replication of the results for the numerical and categorical datasets was successful, as we obtained very similar values to the original paper. However, it turned out to be more challenging to replicate the results for text clustering, mainly due to the limited specification Modha and Spangler (2003) provide regarding the preprocessing method, and the values we obtained were substantially different from the original values. Upon extending the algorithm, we found that the inclusion of the three additional distance functions increased the micro-p values by a substantial margin for the datasets containing numerical and categorical data, especially for the Heart Disease and Australian datasets. However, the Jaccard distance underperformed in text clustering, yielding much lower values of micro-p compared to the cosine distance. In conclusion, higher micro-p values are certainly attainable through the inclusion of additional distortion measures, as can be seen in the convincing results we attained for the Manhattan, Chebyshev and Jaccard distances for numerical and categorical data.

Our findings support the notion that it is useful to take into account multiple distortion functions when working with k-means clustering, as it can give considerable improvements in the quality of the resulting clustering. Further research could be done on the reasons behind the compatibility of certain datasets and distance functions. We see clear differences in the performance of the distance functions for the different datasets, but we have not yet explored the exact reasons why. Because running the k-means algorithm for every combination of distance functions is computationally expensive, it would be very useful to be able to predict beforehand which distance functions are likely to yield the best results. Another interesting topic for research is the cluster initialization process. It is well known that the initially chosen clusters can influence the quality of the clustering greatly. We attempt to take away some of this randomness by using different seeds for the initialization but, once again, this is computationally expensive. Therefore, it would be a very welcome addition if we either a) develop an initialization process which structurally outperforms the random process we employed in this thesis, or b) invent a metric with which the quality of the initial clustering can be assessed, such that we do not have to run the entire algorithm to know which seed will give the best results. In particular, it could be very valuable to invest more time into improving the initialization technique for text clustering with the Jaccard distance. As mentioned in the results, the initialization seems to be a major bottleneck for this clustering. Lastly, another limiting factor for the performance of the convex k-means algorithm is the method through which the optimal feature weighting is determined. The current implementation requires iterating through all combinations of feature weightings. A substantial improvement could be made if we could determine this optimal feature weighting more efficiently, rather than having to run the algorithm for all possible combinations.

References

Abuobieda, A., Salim, N., Binwahlan, M. S. & Osman, A. H. (2013). Differential evolution cluster-based text summarization methods. In *2013 international conference on computing, electrical and electronic engineering (icceee)* (p. 244-248). doi: 10.1109/ICCEEE.2013.6633941

Ahmad, A. & Dey, L. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data Knowledge Engineering*, *63*(2), 503-527. doi: 10.1016/j.datak.2007.03.016

Ahmed, M. H., Tiun, S., Omar, N. & Sani, N. S. (2023). Short text clustering algorithms, application and challenges: A survey. *Applied Sciences*, *13*(1). doi: 10.3390/app13010342

Arthur, D. & Vassilvitskii, S. (2007). k-means plus plus : The advantages of careful seeding. In *Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms* (p. 1027-1035). Retrieved from `https://courses.cs.duke.edu/spring07/cps296.2/papers/kMeansPlusPlus.pdf`

Becker, B. & Kohavi, R. (1996). *Adult.* UCI Machine Learning Repository. doi: 10.24432/C5XW20

Bianchi, F., Terragni, S. & Hovy, D. (2020). Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. *arXiv preprint arXiv:2004.03974*. doi: 10.48550/arXiv.2004.03974

Chan, E. Y., Ching, W. K., Ng, M. K. & Huang, J. Z. (2004). An optimization algorithm for clustering using weighted dissimilarity measures. *Pattern Recognition*, *37*(5), 943-952. doi: 10.1016/j.patcog.2003.11.003

Cordeiro de Amorim, R. & Mirkin, B. (2012). Minkowski metric, feature weighting and anomalous cluster initializing in k-means clustering. *Pattern Recognition*, *45*(3), 1061-1075. doi: 10.1016/j.patcog.2011.08.012

de Amorim, R. C. (2016, JUL). A survey on feature weighting based k-means algorithms. *JOURNAL OF CLASSIFICATION*, *33*(2), 210-242. doi: 10.1007/s00357-016-9208-4

Dhillon, I. & Modha, D. (2001, JAN). Concept decompositions for large sparse text data using clustering. *MACHINE LEARNING*, *42*(1-2), 143-175. doi: 10.1023/A:1007612920971

Dorman, K. S. & Maitra, R. (2022). An efficient k-modes algorithm for clustering categorical datasets. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*(1), 83-97. doi: 10.1002/sam.11546

Ezugwu, A. E., Ikotun, A. M., Oyelade, O. O., Abualigah, L., Agushaka, J. O., Eke, C. I. & Akinyelu, A. A. (2022). A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, *110*, 104743. doi: 10.1016/j.engappai.2022.104743

Frankes, W. B. & Baeza-Yates, R. (1992). Information retrieval: Data structure & algorithms.

Hornik, K., Feinerer, I., Kober, M. & Buchta, C. (2012). Spherical k-means clustering. *Journal of Statistical Software*, *50*(10), 1–22. doi: 10.18637/jss.v050.i10

Janosi, S. W. P. M., Andras & Detrano, R. (1988). *Heart Disease.* UCI Machine Learning Repository. doi: 10.24432/C52P4X

Kongsin, T. & Klongboonjit, S. (2020). Machine component clustering with mixing technique of dsm, jaccard distance coefficient and k-means algorithm. In *2020 ieee 7th international conference on industrial engineering and applications (iciea 2020)* (p. 251-255). doi: 10.1109/iciea49774.2020.9101912

Leisch, F. (2006). A toolbox for k-centroids cluster analysis. *Computational Statistics Data Analysis*, *51*(2), 526-544. doi: 10.1016/j.csda.2005.10.006

Levandowsky, M. & Winter, D. (1971). Distance between sets. *Nature*, *234*(5323), 34-35. doi: 10.1038/234034a0

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, *5.1*, 281-297.

Mitchell, T. (1999). *Twenty Newsgroups.* UCI Machine Learning Repository. doi: 10.24432/C5C323

Modha, D. & Spangler, W. (2003, SEP). Feature weighting in k-means clustering. *MACHINE LEARNING*, *52*(3), 217-237. doi: 10.1023/A:1024016609528

Morissette, L. & Chartier, S. (2013). The k-means clustering technique: General considerations and implementation in mathematica. *Tutorials in Quantitative Methods for Psychology*, *9*(1), 15-24. doi: 10.20982/tqmp.09.1.p015

Oyelade, J., Isewon, I., Oladipupo, O., Emebo, O., Omogbadegun, Z., Aromolaran, O., ... Olawole, O. (2019). Data clustering: Algorithms and its applications. In S. Misra et al. (Eds.), *2019 19th international conference on computational science and its applications (iccsa 2019)* (p. 71-81). doi: 10.1109/ICCSA.2019.000-1

Pandit, S., Gupta, S. et al. (2011). A comparative study on distance measuring approaches for clustering. *International journal of research in computer science*, *2*(1), 29–31. doi: 10.7815/ijorcs.21.2011.011

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. Retrieved from `https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html`

Quinlan, R. (n.d.). *Statlog (Australian Credit Approval).* UCI Machine Learning Repository. doi: 10.24432/C59012

Schubert, E., Lang, A. & Feher, G. (2021). Accelerating spherical k-means. In *International conference on similarity search and applications* (pp. 217–231). doi: 10.1007/978-3-030-89657-7_17

Sculley, D. (2010). Web-scale k-means clustering. In *Proceedings of the 19th international conference on world wide web* (p. 1177–1178). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/1772690.1772862

Sinaga, K. P. & Yang, M.-S. (2020). Unsupervised k-means clustering algorithm. *IEEE Access*, *8*, 80716-80727. doi: 10.1109/ACCESS.2020.2988796

Yin, J. & Wang, J. (2016). A text clustering algorithm using an online clustering scheme for initialization. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1995–2004). doi: 0.1145/2939672.2939841

A Programming code

We implement the convex k-means algorithm in Python. The code takes the following parameters as input:

- The numerical distance functions to apply
- The categorical distance functions to apply
- The datasets to use
- The numbers of clusters
- The number of seeds to use
- A boolean stating whether we are looking for an optimal feature weighting or performing the algorithm for a fixed feature weighting
- A boolean stating whether we wish to generate a plot

The code consists of three classes: main.py, datasets.py and convexKMeans.py. We use main.py to set the input parameters and set up the framework for running the algorithm. Datasets.py contains functions to prepare the datasets for clustering. ConvexKMeans.py contains the implementation of the algorithm and the majority of the code is found in this class. A ZIP file containing our code and an explanation as to how to replicate our results is available.

B Additional results

Table 9 shows the replication results for a fixed feature weighting for the datasets Adult and Australian. Tables 10, 11 and 12 show the results for the optimal feature weighting for all combinations of distance functions for the datasets Heart Disease, Adult and Australian, respectively.

Table 9: Original and replicated results for the datasets Adult and Australian for a fixed feature weighting

		Original results		Replicated results	
k	$\tilde{\alpha}$	$\mathcal{Q}(\tilde{\alpha})$	micro-p	$\mathcal{Q}(\tilde{\alpha})$	micro-p
		Adult			
2	(.5, .5)	154.44	.759	150.85	.759
4	(.5, .5)	24.80	.769	35.94	.769
6	(.5, .5)	13.68	.761	13.39	.765
8	(.5, .5)	10.49	.770	8.87	.769
16	(.5, .5)	2.68	.800	3.03	.788
		Australian			
2	(.5, .5)	107.29	.646	103.64	.650
4	(.5, .5)	30.16	.648	34.10	.648
6	(.5, .5)	13.35	.686	14.96	.668
8	(.5, .5)	10.75	.690	7.74	.694
16	(.5, .5)	2.75	.738	2.79	.730

Table 10: Results for the optimal feature weighting for all combinations of distance functions for the dataset Heart Disease

micro-p	macro-p	macro-r	$\alpha^{\dagger}$	Numerical distance function	Categorical distance function
			$k = 2$		
.791	.790	.789	(.16, .84)	Manhattan	Cosine
.801	.801	.799	(.09, .91)	Euclidean	Cosine
.818	.823	.813	(.09, .91)	Chebyshev	Cosine
.779	.779	.775	(.20, .80)	Manhattan	Jaccard
.742	.741	.739	(.14, .86)	Euclidean	Jaccard
.817	.827	.809	(.01, .99)	Chebyshev	Jaccard
			$k = 4$		
.771	.795	.762	(.19, .81)	Manhattan	Cosine
.797	.797	.796	(.09, .91)	Euclidean	Cosine
.804	.815	.802	(.16, .84)	Chebyshev	Cosine
.779	.805	.769	(.16, .84)	Manhattan	Jaccard
.779	.803	.768	(.11, .89)	Euclidean	Jaccard
.814	.823	.808	(.18, .82)	Chebyshev	Jaccard
			$k = 6$		
.774	.780	.769	(.18, .82)	Manhattan	Cosine
.788	.805	.779	(.09, .91)	Euclidean	Cosine
.812	.817	.807	(.23, .77)	Chebyshev	Cosine
.766	.778	.760	(.18, .82)	Manhattan	Jaccard
.766	.777	.758	(.13, .87)	Euclidean	Jaccard
.813	.821	.807	(.10, .90)	Chebyshev	Jaccard
			$k = 8$		
.780	.792	.772	(.15, .85)	Manhattan	Cosine
.795	.805	.790	(.11, .89)	Euclidean	Cosine
.798	.799	.794	(.20, .80)	Chebyshev	Cosine
.790	.800	.783	(.15, .85)	Manhattan	Jaccard
.778	.784	.773	(.13, .87)	Euclidean	Jaccard
.809	.815	.803	(.15, .85)	Chebyshev	Jaccard
			$k = 16$		
.795	.803	.790	(.18, .82)	Manhattan	Cosine
.788	.791	.785	(.15, .85)	Euclidean	Cosine
.803	.805	.801	(.16, .84)	Chebyshev	Cosine
.806	.808	.803	(.16, .84)	Manhattan	Jaccard
.782	.784	.779	(.13, .87)	Euclidean	Jaccard
.813	.814	.810	(.20, .80)	Chebyshev	Jaccard

Table 11: Results for the optimal feature weighting for all combinations of distance functions for the dataset Adult

micro-p	macro-p	macro-r	$\alpha^\dagger$	Numerical distance function	Categorical distance function
			$k = 2$		
.759	.380	.500	(.26, .74)	Manhattan	Cosine
.759	.380	.500	(.15, .85)	Euclidean	Cosine
.772	.608	.548	(.29, .71)	Chebyshev	Cosine
.759	.380	.500	(.16, .84)	Manhattan	Jaccard
.759	.380	.500	(.14, .86)	Euclidean	Jaccard
.759	.380	.500	(.10, .90)	Chebyshev	Jaccard
			$k = 4$		
.763	.503	.576	(.25, .75)	Manhattan	Cosine
.799	.723	.679	(.12, .88)	Euclidean	Cosine
.780	.850	.556	(.15, .85)	Chebyshev	Cosine
.762	.500	.575	(.20, .80)	Manhattan	Jaccard
.763	.555	.527	(.09, .91)	Euclidean	Jaccard
.763	.678	.508	(.08, .92)	Chebyshev	Jaccard
			$k = 6$		
.810	.750	.686	(.18, .82)	Manhattan	Cosine
.786	.710	.665	(.13, .87)	Euclidean	Cosine
.771	.790	.534	(.24, .76)	Chebyshev	Cosine
.770	.681	.658	(.22, .78)	Manhattan	Jaccard
.768	.720	.573	(.10, .90)	Euclidean	Jaccard
.765	.862	.513	(.09, .91)	Chebyshev	Jaccard
			$k = 8$		
.811	.756	.681	(.19, .81)	Manhattan	Cosine
.805	.736	.700	(.14, .86)	Euclidean	Cosine
.768	.839	.520	(.16, .84)	Chebyshev	Cosine
.776	.635	.651	(.22, .78)	Manhattan	Jaccard
.770	.680	.578	(.14, .86)	Euclidean	Jaccard
.766	.858	.514	(.05, .95)	Chebyshev	Jaccard
			$k = 16$		
.812	.760	.676	(.23, .77)	Manhattan	Cosine
.808	.747	.676	(.15, .85)	Euclidean	Cosine
.784	.764	.575	(.19, .81)	Chebyshev	Cosine
.792	.714	.676	(.16, .84)	Manhattan	Jaccard
.784	.703	.629	(.14, .86)	Euclidean	Jaccard
.773	.780	.536	(.12, .88)	Chebyshev	Jaccard

Table 12: Results for the optimal feature weighting for all combinations of distance functions for the dataset Australian

micro-p	macro-p	macro-r	$\alpha^{\dagger}$	Numerical distance function	Categorical distance function
			$k = 2$		
.818	.830	.806	(.20, .80)	Manhattan	Cosine
.821	.827	.811	(.10, .90)	Euclidean	Cosine
.758	.714	.756	(.01, .99)	Chebyshev	Cosine
.775	.778	.766	(.20, .80)	Manhattan	Jaccard
.789	.800	.776	(.08, .92)	Euclidean	Jaccard
.794	.792	.795	(.00, 1.00)	Chebyshev	Jaccard
			$k = 4$		
.772	.782	.763	(.19, .81)	Manhattan	Cosine
.789	.794	.779	(.09, .91)	Euclidean	Cosine
.798	.805	.798	(.02, .98)	Chebyshev	Cosine
.733	.736	.720	(.20, .80)	Manhattan	Jaccard
.751	.759	.744	(.11, .89)	Euclidean	Jaccard
.806	.815	.803	(.03, .97)	Chebyshev	Jaccard
			$k = 6$		
.770	.792	.753	(.20, .80)	Manhattan	Cosine
.804	.810	.797	(.10, .90)	Euclidean	Cosine
.798	.815	.793	(.03, .97)	Chebyshev	Cosine
.766	.789	.749	(.17, .83)	Manhattan	Jaccard
.730	.759	.713	(.13, .87)	Euclidean	Jaccard
.803	.808	.800	(.01, .99)	Chebyshev	Jaccard
			$k = 8$		
.802	.810	.792	(.17, .83)	Manhattan	Cosine
.756	.781	.738	(.16, .84)	Euclidean	Cosine
.808	.816	.804	(.03, .97)	Chebyshev	Cosine
.794	.798	.793	(.00, 1.00)	Manhattan	Jaccard
.765	.779	.751	(.10, .90)	Euclidean	Jaccard
.795	.798	.784	(.05, .95)	Chebyshev	Jaccard
			$k = 16$		
.818	.820	.814	(.16, .84)	Manhattan	Cosine
.797	.807	.784	(.12, .88)	Euclidean	Cosine
.834	.836	.836	(.03, .97)	Chebyshev	Cosine
.803	.804	.797	(.16, .84)	Manhattan	Jaccard
.794	.805	.782	(.11, .89)	Euclidean	Jaccard
.813	.814	.810	(.00, 1.00)	Chebyshev	Jaccard