Central Clustering of Categorical Data with Automated Feature Weighting

Lifei Chen  
Fujian Normal University  
China  
clfei@fjnu.edu.cn

Shengrui Wang  
University of Sherbrooke  
Canada  
shengrui.wang@usherbrooke.ca

Abstract

The ability to cluster high-dimensional categorical data is essential for many machine learning applications such as bioinformatics. Currently, central clustering of categorical data is a difficult problem due to the lack of a geometrically interpretable definition of a cluster center. In this paper, we propose a novel kernel-density-based definition using a Bayes-type probability estimator. Then, a new algorithm called $k$-centers is proposed for central clustering of categorical data, incorporating a new feature weighting scheme by which each attribute is automatically assigned with a weight measuring its individual contribution for the clusters. Experimental results on real-world data show outstanding performance of the proposed algorithm, especially in recognizing the biological patterns in DNA sequences.

1 Introduction

Clustering high-dimensional categorical data with attribute weighting is an essential process in many machine learning applications. For example, for DNA sequences consisting of nucleotides encoded in one of the four categories A, G, T and C, such a clustering algorithm can be used to reveal hidden biological concepts (clusters) and to assess their degree of interest by the corresponding attribute weights. Compared to numeric data, for which numerous clustering methods have been developed [Jain et al., 1999], categorical data pose a unique challenge in clustering tasks, due to the difference between the two data types.

Unlike the numeric case, when the data are categorical the set mean is an undefined concept. This means that the well-known $k$-means [Jain et al., 1999] and its numerous variants [Huang et al., 2005; Jing et al., 2007; Chen et al., 2012] cannot be directly used for center-based clustering, alternatively known as central clustering, of categorical data. Owing to its strength in geometrical interpretation and cluster representation and its computational efficiency, central clustering is one of the mainstream methods in the machine learning community. Due to the nature of discrete space and the consequent lack of a “mean” concept, one has to resort to the mode [Huang and Ng, 2003; Chan et al., 2004; Bai et al., 2011] for representing the “center” of a categorical cluster. Statistically speaking, this approach can only capture partial information on the data objects in a cluster. Recently, a few attempts have been made to define cluster centers as extensions to the mode: for example, $k$-representatives [San et al., 2004] suggests a frequency estimator for the definition. However, such an estimator typically results in large estimation variance, as measured by the finite-sample mean squared error [Ouyang et al., 2006; Li and Racine, 2007].

Another challenging issue is attribute weighting, which entails automatically identifying the individual contributions of attributes for the clusters. Though automated attribute weighting has been stressed extensively in numeric data clustering [Huang et al., 2005; Lu et al., 2011], few attempts have been made to apply it to categorical data clustering. The main obstacle lies in the difficulty of estimating the attribute weights based on the statistics of categories in a cluster. Those measures that have been successfully used for weighting numeric attributes, such as the popular variance of numeric data, are not well-defined for categorical data [Light and Marglin, 1971]. In fact, in the existing methods [Chan et al., 2004; Bai et al., 2011; Xiong et al., 2012], an attribute is weighted solely according to the mode category for that attribute. Consequently, the weights easily yield a biased indication of the importance of attributes to clusters.

In this paper, we define the mean of a categorical data set as a statistical center of the set, estimated by a kernel density estimation method. In effect, the cluster center is a Bayes-type probability estimator that has the frequency estimator [San et al., 2004; Kim et al., 2005] as a special case. For the $k$-centers algorithm proposed in this paper, the new formulation of the categorical cluster center is used to derive a weight calculation expression that correlates with the average deviation of categories in a cluster. This is consistent with the method commonly used to weight a numeric attribute [Huang et al., 2005]. We conducted a series of experiments on real-world categorical data. The results show that $k$-centers significantly outperforms other mainstream clustering algorithms especially for the task of recognizing biological concepts in DNA sequences.

The remainder of this paper is organized as follows: Section 2 defines the probabilistic center for a categorical data set, and presents a kernel density estimation method for the
probability estimation. In Section 3, the k-centers algorithm is presented. Section 4 describes related work. Experimental results are presented in Section 5. Finally, Section 6 gives our conclusion and discusses directions for future work.

2 Probabilistic Center of a Categorical Cluster

The aim of this section is to propose a geometrically interpretable definition for cluster centers in categorical data. We begin by introducing the notation used throughout the paper. In what follows, the dataset is denoted by 

\[ DB = \{x_1, x_2, \ldots, x_N\} \]

from which \( k \) (\( 1 < k < N \)) clusters are searched for. Here \( x_i = (x_{i1}, x_{i2}, \ldots, x_{iD}) \) for \( i = 1, 2, \ldots, N \) are data objects. For the \( d \)th categorical attribute, where \( d = 1, 2, \ldots, D \), we denote the set of categories by \( O_d \); i.e., the \( d \)th attribute takes \( |O_d| \) (> 1) discrete values. To unambiguously identify the categories in \( O_d \), we suppose that each of them is assigned a unique index \( l \), where \( l \in [1, |O_d|] \), and denote the \( l \)th category by \( o_{dl} \in O_d \). Moreover, the \( k \) clusters are denoted by \( c_1, c_2, \ldots, c_k \), each consisting of a disjoint subset of \( DB \); therefore, \( DB = \bigcup_{j=1}^{k} c_j \). The number of data objects in the \( j \)th cluster is denoted by \( n_j \), and the set of \( k \) clusters by \( C = \{ c_j \}_{j=1}^{k} \).

In central clustering methods, each cluster is represented by its “center”. For example, in the popular \( k \)-means method for clustering numeric data, the cluster center is defined as the mean of the data objects in that cluster. Obviously, such a definition cannot be directly used for categorical clusters, because the concept of mean is meaningless for categorical data, where each attribute can only take a discrete value. In fact, from a statistical perspective, the cluster center of a numeric cluster is exactly the expectation of a continuous random variable associated with the data, implicitly based on the assumption that the variable follows a Gaussian distribution. We therefore propose a generalized definition for categorical clusters, by denoting the center of \( c_j \) as \( v_j = \{v_{jd}\}_{d=1}^{D} \), with the \( d \)th element being a vector in the probability space, where \( O_d \) serves as the sample space and \( P_{jd} \) as the probability measure defined on the Borel set of the sample space with regard to data subset \( c_j \).

**Definition 1.** The probabilistic center of \( c_j \) on the \( d \)th dimension is 

\[ v_{jd} = (P_{jd}(o_{d1}), \ldots, P_{jd}(o_{dl}), \ldots, P_{jd}(o_{dlO_d})) > . \]

One of the implementations of \( v_{jd} \) is the frequency estimator, where each element \( P_{jd}(o_{dl}) \) is estimated by 

\[ f_j(o_{dl}) = \frac{\#_{j}(o_{dl})}{n_j} \]

with \( \#_{j}(o_{dl}) \) being the number of \( o_{dl} \) appearing in \( c_j \). Such an estimator typically has the least sample bias; at the same time, however, it may also have a large estimation variance, in terms of the finite-sample mean squared error [Ouyang et al., 2006; Li and Racine, 2007]. Actually, the approximation of \( f_j(o_{dl}) \) for \( P_{jd}(o_{dl}) \) holds only if the cluster size (say, \( n_j \)) is infinitively large. This is unrealistic in real-world applications: for example, in the task of recognizing biological concepts in DNA sequences [Noordewier et al., 1991], the number of samples is typically small. To obtain an optimal estimation in terms of trade-off between sample bias and estimation variance, in the work described here, we employ the kernel smoothing method for the probability estimation, as follows:

Let \( X_d \) be a random variable associated with the observations \( x_{id} \) for \( i = 1, 2, \ldots, n_j \), and denote the probability density by \( p(X_d) \). Using a kernel density estimation method (KDE), \( p(X_d) \) is defined on the kernel function, given by \( \ell(X_d, o_{dl}, \lambda) \). Here, \( o_{dl} \in O_d \) for \( l = 1, 2, \ldots, |O_d| \) and \( \lambda \) is the smoothing parameter called bandwidth. We use a variation on Aitchison & Aitken’s kernel function [Aitchison and Aitken, 1976] defined by 

\[ \ell(X_d, o_{dl}, \lambda_j) = \begin{cases} \frac{1}{|O_d|} - \lambda_j & X_d = o_{dl} \\ \frac{1}{|O_d|} & X_d \neq o_{dl} \end{cases} \]

with \( \lambda_j \in [0, 1] \) being the unique bandwidth for \( c_j \). Letting \( \hat{p}(X_d | \lambda_j) \) be the kernel estimator of \( p(X_d) \), we have 

\[ \hat{p}(X_d | \lambda_j) = \frac{1}{n_j} \sum_{m=1}^{n_j} \ell(X_d, x_{imd}, \lambda_j) 
\]

\[ = f_j(X_d) + \left( \frac{1}{|O_d|} - f_j(X_d) \right) \lambda_j . \]

Then, the probabilistic center of a categorical cluster in Definition 1 can be estimated based on Eq. (2), yielding 

\[ \hat{P}_{jd}(o_{dl}) = \hat{p}(o_{dl} | \lambda_j) = \lambda_j \frac{1}{|O_d|} + (1 - \lambda_j) f_j(o_{dl}). \]

Note that Eq. (3) can be viewed as a Bayes-type probability estimator [Ouyang et al., 2006], since it is a weighted average of a uniform probability (the first term \( \frac{1}{|O_d|} \)) as a prior, and a frequency estimator (the second term \( f_j(o_{dl}) \)) as the posterior. When the bandwidth \( \lambda_j = 1 \), \( \hat{P}_{jd}(o_{dl}) \) degenerates to a uniform distribution. In this case, the categories in the \( d \)th attribute are “smoothed out”. In the opposite case where \( \lambda_j = 0 \), the center degenerates to the pure frequency estimator, which is the case in [San et al., 2004] as well as [Kim et al., 2005] where the frequency estimator is directly used to represent the “mean” of a categorical dataset.

The selection of bandwidth is an important issue for such a KDE method, because it is the value of bandwidth that dominates the probability distribution for a given data set. For the purpose, we employ the least squares cross-validation (LSCV), an automatic data-driven method that is widely used for optimal bandwidth selection [Li and Racine, 2007]. The LSCV method is based on the principle of selecting a bandwidth that minimizes the total error of the resulting estimation, over all the data objects, i.e., \( \phi(\lambda_j) = \sum_{d=1}^{D} \sum_{o \in O_d} [\hat{p}(o | \lambda_j) - p(o)]^2 \). The optimal \( \lambda_j \) that minimizes \( \phi(\lambda_j) \), denoted as \( \lambda_j^* \), is determined in the following Proposition 1.

**Proposition 1.** Given \( n_j \) inputs of \( c_j \), the optimal bandwidth is

\[ \lambda_j^* = \frac{1}{n_j - 1} \sum_{d=1}^{D} \frac{\sum_{o \in O_d} [f_j(o)]^2 - f_j(o_{dl})^2}{|O_d|} \]

in the sense of the least squares cross-validation.
Proof. First, the objective function can be rewritten as
\[
\phi_1(\lambda_j) = \sum_{d=1}^{D} \sum_{a \in O_d} \frac{1}{|a|} [p(o|\lambda_j)]^2 - 2 \sum_{d=1}^{D} \sum_{a \in O_d} p(o) \hat{p}(o|\lambda_j) \text{, with removal of the constant}
\]
\[
\sum_{d=1}^{D} \sum_{a \in O_d} [p(o)]^2.
\]
Note that the term \(\sum_{o \in O_d} p(o) \hat{p}(o|\lambda_j)\) is the expectation of \(X_d^2\), therefore, it can be estimated by the sample mean over all the observations. Following [Ouyang et al., 2006], we replace the term with \(\frac{1}{n_d} \sum_{y_{id} \in O_d} \hat{p}(x_{id} | \lambda_j)\), where \(\hat{p}(x_{id} | \lambda_j) = \frac{1}{n_d} \sum_{y_{id} \in O_d} \hat{f}(x_{id})\) is the leave-one-out kernel estimator. Then, the objective function becomes
\[
\phi_2(\lambda_j) = \sum_{d=1}^{D} \sum_{o \in O_d} \frac{1}{|a|} [p(o|\lambda_j)]^2 - 2 \sum_{y_{id} \in O_d} \hat{f}(x_{id}) \hat{p}(o|\lambda_j) + \frac{O_d}{|O_d|} \lambda_j - 1 \). Setting \(\frac{\partial \phi_2}{\partial \lambda_j} = 0\), Eq. (4) follows.

\[ \]

3 \ k\text{-Centers Clustering}

In this section, a central clustering algorithm called \(k\)-centers is proposed to group a categorical dataset DB into \(k\) clusters. As with the \(k\)-means [Jain et al., 1999] designed for numeric clustering, a set of centers \(V = \{v_j\}_{j=1}^{k}\) is used to characterize the \(k\) clusters. We will begin by defining the clustering criterion that needs to be optimized by \(k\)-centers.

3.1 Clustering Criterion

To measure the dissimilarity between a data object and its center, we first express each data object \(x_i\) by a set of vectors \(\{y_{id}\}_{d=1}^{D}\), with \(y_{id} = I(x_{id} = a_{d1}) \ldots I(x_{id} = a_{dk})\). Here \(I()\) is an indicator function whose value is either 1 or 0, indicating whether \(x_{id}\) is the same as \(a_{d}\) in \(O_d\) or not. Then, the dissimilarity on the \(d\)th dimension can be measured by
\[
DIS_d(x_i, v_j) = ||y_{id} - v_{jd}||^2
\]
where the Euclidean norm \(||a||^2\) is given by \(\sqrt{\sum_{d=1}^{D} a_d^2}\).

To weight attributes according to their individual contributions in clustering, like the scheme used in numeric data clustering [Lu et al., 2011; Chen et al., 2012], we introduce a weight vector \(< w_1, w_2, \ldots, w_D >\), satisfying
\[
\begin{align*}
\sum_{d=1}^{D} w_{jd} &= 1, \\
0 &< w_{jd} < 1, \\
&j = 1, 2, \ldots, k; d = 1, 2, \ldots, D
\end{align*}
\]
for cluster \(j\). Intuitively, the weight \(w_{jd}\) is defined to measure the relevance of the \(d\)th attribute to \(c_j\). The greater the relevance, the higher the weight. Based on these definitions, the clustering algorithm should minimize
\[
J(C, W) = \sum_{j=1}^{k} \frac{1}{n_j} \sum_{x_i \in c_j} \sum_{d=1}^{D} w_{jd} [DIS_d(x_i, v_j)]^2
\]
where \(W = \{w_{jd}\}_{k \times D}\) is the weight matrix.

Due to the inclusion of \(w_{jd}\) in \(J(C, W)\), the objective function is non-convex. A common method for convexifying the objective is to add a \(\log \) smoothing term, alternatively known as an entropy term in entropy-based clustering [Jing et al., 2007], which serves to push the minimum of the objective away from the discrete points. In this way, the objective function can be obtained as
\[
J(C, W) = J_1(C, W) + \sum_{j=1}^{k} \xi_j (1 - \sum_{d=1}^{D} w_{jd}) + \beta \sum_{j=1}^{k} \sum_{d=1}^{D} w_{jd} \log(w_{jd})
\]
where the parameter \(\beta (> 0)\) controls the degree of convexity; and \(\xi_j\) for \(j = 1, 2, \ldots, k\) are the Lagrange multipliers enforcing the constraints of Eq. (5).

3.2 Clustering Algorithm

Given DB to be clustered into \(k\) clusters of \(C\), the goal is to look for a saddle point by minimizing \(J(C, W)\) with respect to \(C\) and \(W\), and maximizing with respect to the Lagrange multipliers \(\xi_j\) for \(j = 1, 2, \ldots, k\). The usual method of achieving this is to use the partial optimization for each parameter. Following this method, minimization of \(J(C, W)\) can be performed by optimizing \(C\) and \(W\) in a sequential structure analogous to the mathematics of the EM algorithm [Xu and Jordan, 1996]. In each iteration, we first set \(W = W\), and solve \(C\) as \(\hat{C}\) to minimize \(J(C, W)\). Then, \(C = \hat{C}\) is set and the optimal \(W\), say \(\hat{W}\), is solved to minimize \(J(C, \hat{W})\). The first problem can be solved by assigning each input \(x_i\) to its most similar center in terms of the values of the weighted Euclidean norm. Formally, we assign \(x_i\) to cluster \(m\) according to
\[
m = \arg\min_{v_j} \frac{1}{n_j} \sum_{d=1}^{D} \hat{w}_{jd} \times [DIS_d(x_i, v_j)]^2.
\]

The second optimization problem is solved according to the following proposition:

Proposition 2. Set \(C = \hat{C}\). \(J(\hat{C}, W)\) is minimized iff
\[
\hat{w}_{jd} = \frac{\hat{w}_{jd}}{\sum_{d=1}^{D} \hat{w}_{jd}}
\]
with
\[
-\beta \times \log(\hat{w}_{jd}) = 1 - \frac{\lambda_j^2}{|O_d|} + (\lambda_j^2 - 1) \sum_{o \in O_d} [f_j(o)]^2
\]
for \(j = 1, 2, \ldots, k\) and \(d = 1, 2, \ldots, D\).

Proof. By setting \(\frac{\partial J(C, W)}{\partial w_{jd}} = 0\) and \(\frac{\partial J(C, W)}{\partial \lambda_j} = 0\) for \(d = 1, 2, \ldots, D\) and \(j = 1, 2, \ldots, k\), the results follow.

It can be seen that the new weighting scheme weights a categorical attribute according to the average deviation of the categories on that dimension. In fact, the right side of Eq. (9) is precisely the Gini Index (GI for short) [Sen, 2005] when \(\lambda_j = 0\). An attribute with small dispersion will receive a high weight, indicating that the dimension is more important than others in forming the cluster. Note that this dispersion-based weighting scheme is consistent with the one used for numeric data clustering [Huang et al., 2005], where virtually all of the existing methods compute feature weights as being inversely proportional to the dispersion of the numeric values from the mean in the dimension of the cluster.

The \(k\)-centers algorithm, as outlined by Algorithm 1, performs central clustering on categorical data using the optimization methods presented above. In terms of algorithmic
structure, $k$-centers can be viewed as an extension to the EM algorithm [Xu and Jordan, 1996]. Therefore, we refer the reader to that paper for the discussions of convergence. The computational complexity of $k$-centers is $O(kNDM)$, where $M$ denotes the number of iterations.

### 4 Related Work

Existing categorical clustering algorithms fall into two groups according to whether a clustering objective function is explicitly defined for the clustering process. Hierarchical clustering algorithms, which organize data objects into a tree of clusters, are representatives of the first group: a clustering objective function is not necessary in these algorithms. Examples include ROCK [Guha et al., 2000] and the recently published DHCC [Xiong et al., 2012]. Generally, algorithms in this group have a high time complexity, reaching $O(N^2\log N)$. The goal of the algorithms in the second group is to seek an optimum grouping of the data objects by optimizing a specially designed clustering criterion, generally defined on partitioning entropy [Li et al., 2004] or directly on category frequency [Gan et al., 2006; Cesario et al., 2007].

In these algorithms, often, a Monte-Carlo type optimization method is used to search for a suboptimal solution of the objective function. Typically, a large number of iterations is needed using such a method.

Inspired by the success of $k$-means (for central clustering of numeric data) [Jain et al., 1999], a number of $k$-means-type clustering objective functions have been defined for the second group. The advantages of such an objective function include the possibility of geometrical interpretation and the feasibility of using a more efficient optimization method, such as the popular EM [Xu and Jordan, 1996]. The $k$-modes [Huang and Ng, 2003] represents the “mean” of a categorical cluster by the mode category, while [Lee and Pedrycz, 2009] uses a fuzzy p-mode prototype. In addition, $k$-populations [Kim et al., 2005] and $k$-representatives [San et al., 2004] define their cluster centers based on the frequency estimators, which can be viewed as a non-smoothed implementation of our probabilistic center defined in Eq. (3).

For a clustering task, attribute weighting is commonly performed by assigning a weighting value to each attribute during the clustering process [Huang et al., 2005; Lu et al., 2011; Chen et al., 2012]. Existing attribute-weighting schemes can be roughly divided into two groups. In one group, each weight is computed according to the average distance of data objects from the mode of a cluster: one example is WKM [Chan et al., 2004]. In the second group, which includes such algorithms as MWKM [Bai et al., 2011] and DHCC [Xiong et al., 2012], weights are computed based on the frequency of the mode category. It can be seen that only partial sections of the category distribution are considered in these weighting schemes.

### 5 Experimental Evaluation

Below, we evaluate the performance of $k$-centers on real-world categorical datasets, and we also experimentally compare $k$-centers with some mainstream clustering algorithms.

#### 5.1 A Case Study

This set of experiments aims at examining the performance of $k$-centers by a case study on DNA sequence data. The task is to recognize promoters in DNA sequences, where a promoter is a genetic region which initiates the first step in the expression of an adjacent gene [Towell et al., 1990]. The database used was the well-known *E. Coli* promoter gene sequences [Harley and Reynolds, 1987], available at the UCI Machine Learning Repository. The dataset contains 106 samples (53 promoter and 53 non-promoter) featuring 57 attributes, each associated with a nucleotide (A, G, T or C) starting at position -50 (p-50) and ending at position +7 (p7). In [Towell et al., 1990], the promoters in these data falling into two subcategories (namely, *contact* and *conformation* regions) were reported. We shall evaluate our method using this domain theory.

In applying $k$-centers to such a machine learning task, the parameter $k$ is easily set, i.e., $k=2$. Another parameter in the algorithm is $\beta$, which is used to control the degree of convexity of the objective function (Eq. (6)); the value of $\beta$ thus dominates the weight distribution computed for all the attributes to some extent. We examined the relationship of its value to the resulting average clustering quality, and observed that $k$-centers was robust with $\beta \geq 1.5$. Below, we report the detailed clustering results yielded by setting $\beta = 1.5$ based on this observation. The average performance will be evaluated in the next subsection; here, the best results (in terms of *F-score* [Jing et al., 2007]) during the 100 random clusterings are reported.

Figure 1 shows the weight distributions for the attributes of each class yielded by $k$-centers in the clustering results. The x-axis is the DNA nucleotides, sorted by their positions, and the y-axis indicates the weight. As expected, for the non-promoter class, there is no significant change in the weights over all positions. However, for the promoter class, we can see three slices of successive DNA nucleotides that are assigned obviously high weights compared to those for the non-promoter class. According to the domain theory [Towell et al., 1990], the three slices (from the left side
The cluster center of each attribute obtained by \( k \)-centers, shown in the column “Probabilistic center” in Table 1, is presented as a probability vector. In effect, this is a smoothed vector consisting of the category. To examine the strength of the smoothing method used to compute the probabilistic centers in \( k \)-centers, we designed a reduced algorithm called \( k \)-centers* for this case study, by removing Step 2 of \( k \)-centers; that is, the bandwidths \( \lambda_1 \) and \( \lambda_2 \) are set to 0 in \( k \)-centers*.

Table 3 shows the clustering results yielded by different algorithms on the E. Coli promoter gene sequences.

Table 3: Clustering accuracy (in terms of F1-measure) of different algorithms with the bandwidths estimated by \( k \)-centers.

<table>
<thead>
<tr>
<th>Class</th>
<th>MWKM ( k )-centers*</th>
<th>( k )-centers</th>
<th>( \lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>promoter</td>
<td>0.7805</td>
<td>0.8214</td>
<td>0.9533</td>
</tr>
<tr>
<td>non-promoter</td>
<td>0.6966</td>
<td>0.8000</td>
<td>0.9524</td>
</tr>
</tbody>
</table>

For the non-promoter class, \( k \)-centers yields a bandwidth as large as 0.9650. This is an expected outcome because the non-promoter class contains negative examples, which means that there is no interesting biological concept (here, the promoters) hidden in the class. With such a large bandwidth, the categories in each attribute would be nearly “smoothed out” (see Eq. (2)), which, in turn, results in a smooth weight curve as Figure. 1 shows. We can see from the results of \( k \)-centers* in Table 3 that, in the case where the bandwidths are set to 0, the clustering quality drops significantly. The table shows similar results with the MWKM algorithm. Note that in \( k \)-centers the bandwidths are estimated adaptively to the clusters. As Eq. (4) shows, one can suppose that the bandwidth is equal to 0 only if there are a very large number of data objects to be clustered. Therefore, for the DNA sequences such as the gene sequences in this case study, a smoothed estimation like the computation of probabilistic centers in \( k \)-centers is virtually necessary for categorical data clustering.

5.2 Performance Comparison

The second set of experiments was designed to compare \( k \)-centers with state-of-the-art algorithms. This comparison of performance was done in terms of average clustering accuracy evaluated on a number of real-world datasets.

Table 4: Details of the real-world datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Dimension ((D))</th>
<th>Classes ((K))</th>
<th>Data size ((N))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breastcancer</td>
<td>9</td>
<td>2</td>
<td>699</td>
</tr>
<tr>
<td>Vote</td>
<td>16</td>
<td>2</td>
<td>435</td>
</tr>
<tr>
<td>Soybean</td>
<td>21</td>
<td>4</td>
<td>47</td>
</tr>
<tr>
<td>Mushroom</td>
<td>21</td>
<td>2</td>
<td>8124</td>
</tr>
<tr>
<td>Promoters</td>
<td>57</td>
<td>2</td>
<td>106</td>
</tr>
<tr>
<td>Splice</td>
<td>60</td>
<td>3</td>
<td>3190</td>
</tr>
</tbody>
</table>

Real-world Datasets

Six widely used categorical datasets were used. Table 4 lists the details. We obtained all six datasets from the UCI Machine Learning Repository. The attributes valued in a single category were removed; and the missing value in each attribute was considered as a special category in our experiments.
The Promoters dataset was used in the previous section for a case study. Here, another DNA database entitled Primate splice-junction gene sequences [Noordewier et al., 1991] (Splice for short) was used. In contrast to the Promoters dataset, the problem posed in Splice is to recognize the boundaries between exons and introns. There are three kinds of boundaries (exon/intron boundaries, intron/exon boundaries and neither), corresponding to the three classes EI, IE and Neither in the dataset. The dataset includes four additional characters D, N, S and R, to indicate ambiguity among the four standard characters A, G, T and C. The reader is referred to [Bai et al., 2011; Xiong et al., 2012] for the detailed descriptions of the other datasets in Table 4, since they have been frequently used in related work.

Experimental Results
Six clustering algorithms, k-centers, k-modes (KM for short) [Huang and Ng, 2003], WKM [Chan et al., 2004], MWKM [Bai et al., 2011] and k-representatives (KR for short) [San et al., 2004] were tested in our experiments. The weighting exponents of WKM and MWKM were set to the author-recommended values 1.8 and 2, respectively. We set $\beta = 1.5$ for k-centers for the reason described in Section 5.1.

The clustering quality was measured in terms of F-score [Jing et al., 2007]. Each dataset in Table 4 was clustered by each algorithm for 100 executions and the average performances are reported in the format average ± standard deviation. This is because all of the algorithms choose their initial cluster centers via random selection methods, and thus the clustering results may vary depending on the initialization. Table 5 illustrates the clustering results, where the best ones are marked in boldface. The table shows that k-centers is able to achieve high-quality overall results, whereas k-modes and WKM perform poorly. All of the competing algorithms encounter difficulties on Promoters and Splice. On these two DNA-sequence datasets, k-centers achieves significant improvements compared to the others.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>k-centers</th>
<th>KR</th>
<th>KM</th>
<th>WKM</th>
<th>MWKM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breastcancer</td>
<td>0.95</td>
<td>0.94</td>
<td>0.81</td>
<td>0.76</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>±0.00</td>
<td>±0.03</td>
<td>±0.15</td>
<td>±0.04</td>
<td>±0.13</td>
</tr>
<tr>
<td>Vote</td>
<td>0.88</td>
<td>0.88</td>
<td>0.86</td>
<td>0.82</td>
<td>0.86</td>
</tr>
<tr>
<td></td>
<td>±0.00</td>
<td>±0.05</td>
<td>±0.01</td>
<td>±0.08</td>
<td>±0.00</td>
</tr>
<tr>
<td>Soybean</td>
<td>0.88</td>
<td>0.86</td>
<td>0.83</td>
<td>0.75</td>
<td>0.87</td>
</tr>
<tr>
<td></td>
<td>±0.13</td>
<td>±0.11</td>
<td>±0.13</td>
<td>±0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>Mushroom</td>
<td>0.78</td>
<td>0.77</td>
<td>0.70</td>
<td>0.67</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>±0.13</td>
<td>±0.15</td>
<td>±0.13</td>
<td>±0.06</td>
<td>±0.14</td>
</tr>
<tr>
<td>Promoters</td>
<td>0.87</td>
<td>0.68</td>
<td>0.60</td>
<td>0.70</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>±0.11</td>
<td>±0.11</td>
<td>±0.08</td>
<td>±0.12</td>
<td>±0.07</td>
</tr>
<tr>
<td>Splice</td>
<td>0.87</td>
<td>0.79</td>
<td>0.41</td>
<td>0.54</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>±0.10</td>
<td>±0.06</td>
<td>±0.02</td>
<td>±0.04</td>
<td>±0.01</td>
</tr>
</tbody>
</table>

Three of the competing algorithms, k-modes, WKM and MWKM, use the mode category of each attribute to represent the cluster “center”. Generally, such methods easily fall into local minima of the clustering objective [Jain et al., 1999], leading to their lower average performances on the real-world datasets. Both WKM and MWKM are extensions of k-modes involving weighting the attributes to identify differences in the importance of attributes in clustering. As correctly pointed out by [Bai et al., 2011], the weighting scheme used in WKM tends to inversely reduce the dissimilarity of samples on an important underlying dimension. This results in its lower accuracy on Breastcancer, Vote, Soybean and Mushroom compared to k-modes. MWKM improves on WKM by weighting the attributes on the frequency of the mode category; therefore, when the number of categories for the attributes becomes large (for example, in the Breastcancer dataset where each attribute takes values from 10 categories), the clustering quality is affected.

k-centers owes its good average performance to optimizing the statistical centers of categorical attributes during the clustering process, which allows weighting attributes based on the overall distribution of categories in a cluster. According to this view, the attribute weighting methods used in WKM and MWKM, which focus on dissimilarity of categories to the mode and the frequency of the mode, respectively, can be regarded as two special cases of our k-centers algorithm. The performances of k-representatives are comparable to that of k-centers on the datasets with relatively low dimensionality. However, like the other competing algorithms, it performs poorly on the DNA sequence data due to the non-smoothing method used for optimizing the cluster centers, and the lack of an adaptive attribute-weighting scheme to distinguish different contributions of attributes to the clusters.

6 Conclusion and Perspectives
In this paper, we first discuss the problem faced by a center-based algorithm in clustering categorical data. This problem becomes difficult due to the fact that general statistical measures such as mean and variance, which are common in numeric data, are undefined for categorical data. We propose a definition for the center of a categorical cluster, called the probabilistic center, by kernel density estimation on categorical data. We also propose a central clustering algorithm called k-centers using the probabilistic center and a built-in feature weighting scheme, which automatically assigns each attribute a weight indicating its individual importance to clusters. The experiments were conducted on six real-world datasets including two common DNA-sequence databases, and the results show its outstanding effectiveness compared with state-of-the-art algorithms. There are many directions that are clearly of interest for future exploration. One avenue of further study is to estimate the parameter $\beta$ adaptively. Another effort will be directed toward extending the algorithm for clustering mixed-type data with both numeric and categorical attributes.

Acknowledgments
The authors are grateful to the anonymous reviewers for their invaluable comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Discovery Accelerator Supplements grant No. 396097-2010, and by the National Natural Science Foundation of China under Grant No. 61175123.
References


