Clustering Categorical Data Based on Distance Vectors

Peng ZHANG, Xiaogang WANG, and Peter X.-K. SONG

We introduce a novel statistical procedure for clustering categorical data based on Hamming distance (HD) vectors. The proposed method is conceptually simple and computationally straightforward, because it does not require any specific statistical models or any convergence criteria. Moreover, unlike most currently existing algorithms that compute the class membership or membership probability for every data point at each iteration, our algorithm sequentially extracts clusters from the given dataset. That is, at each iteration our algorithm strives to identify only one cluster, which will then be deleted from the dataset at the next iteration; this procedure repeats until there are no more significant clusters in the remaining data. Consequently, the number of clusters can be determined automatically by the algorithm. As for the identification and extraction of a cluster, we first locate the cluster center by using a Pearson chi-squared–type statistic on the basis of HD vectors. The partition of the dataset produced by our algorithm is unique and insensitive to the input order of data points. The performance of the proposed algorithm is examined using both simulated and real world datasets. Comparisons with two well-known clustering algorithms, K-modes and AutoClass, show that the proposed algorithm substantially outperforms these competitors, with the classification rate or the information gain typically improved by several orders of magnitude. Computational complexity and run time comparisons are also provided.

KEY WORDS: AutoClass; Categorical data; Clustering; Computational complexity; Distance vector; Hamming distance; K-modes; Modified chi-squared statistic.

1. INTRODUCTION

Clustering is a widely used exploratory tool with applications in business, engineering, and life sciences. The objective of cluster analysis is to discover certain underlying structures of a dataset and classify observations into different subsets, so that associations are high among members of the same subset and low among members from different subsets. The association is typically measured by some pairwise similarity or dissimilarity functions, or simply by a distance function of any two data points. The existing clustering algorithms can be roughly classified into two types, partition algorithms and hierarchical algorithms. An example of the first type is the popular K-means algorithm of MacQueen (1967), which is essentially used to cluster continuous data. In effect, most of the available clustering algorithms in the literature have been developed to deal with continuous data, and very few methods have been proposed specifically for clustering categorical data (see Kaufman and Rousseeuw 1990 and references therein).

Categorical data are pervasive in practice. There are two primary types of scales for categorical variables, nominal and ordinal. In this article we focus on categorical data with nominal scales. The essential difficulty in dealing with nominal categorical data is the lack of a metric space in which data points are positioned with measurable coordinates. An immediate consequence is that those clustering algorithms using the Euclidean or other distance functions for continuous data are not applicable for such categorical data. Ralambondrainy (1995) investigated using the K-means algorithm to cluster categorical data by introducing a numerical coding scheme on categorical attributes through the so-called "dummy variables." For example, binary 0/1 may be used to code the gender attribute. An obvious pitfall of this approach is that it subjectively imposes a metric distance between categorical data points, and such a distance can be rather different when a different coding scheme is used. Clearly, the mean (or arithmetic average) used in the algorithm to represent the cluster center is in fact meaningless. Moreover, an arbitrarily chosen coding scheme may lead to certain false dependence structures among the dummy variables, which would result in a seriously misleading partition of categorical data. To elucidate, let us consider a simple example of two data points with five binary attributes, $\mathbf{x}_1 = [1, 0, 1, 0, 1]$ and $\mathbf{x}_2 = [0, 1, 0, 1, 0]$. In 0/1 coding, these two data points have a perfect -1 correlation coefficient. If we use a different coding scheme for the first attribute as, say $0 \hookrightarrow 2$ or $1 \hookrightarrow 3$, and the original data are changed to $\mathbf{x}'_1 = [3, 0, 1, 0, 1]$ and $\mathbf{x}'_2 = [2, 1, 0, 1, 0]$, then the correlation between $\mathbf{x}'_1$ and $\mathbf{x}'_2$ is now .49, which totally alters the nature and strength of the correlation between the two data points.

To improve the K-means algorithm, Huang (1997) proposed the so-called "K-modes" algorithm, in which modes instead of means are used to represent the cluster centers. Similar to the K-means algorithm, it is an iterative algorithm that updates the membership of each data point and cluster modes at each iteration, and it may rely on user-chosen coding schemes for categorical attributes. To run the K-modes algorithm, one has to provide a value of the parameter K representing the number of clusters. In addition, as pointed out by Huang (1997), the resulting partition of this algorithm is not unique and is heavily dependent on the initial seeds chosen to start the algorithm. Different input orders of the same dataset can produce different clustering results. Therefore, it is certainly appealing to develop a new partition algorithm that can produce more reliable results.

To better appreciate the performance of the K-modes algorithm, we applied it to a well-known categorical dataset, the soybean disease dataset, downloaded from the Machine Learning Depository at the University of California at Irvine. The data used in our analysis comprises of 47 observations, each

Peng Zhang is a Doctoral Student, Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (E-mail: *p5zhang@math.uwaterloo.ca*). Xiaogang (Steven) Wang is Assistant Professor, Department of Mathematics and Statistics, York University, Toronto, Ontario, Canada M3J 1P3 (E-mail: *stevenw@mathstat.yorku.ca*). Peter X.-K. Song is Associate Professor, Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1 (E-mail: *song@uwaterloo.ca*). This research was supported in part by the Natural Sciences and Engineering Research Council of Canada. The authors thank the associate editor and two anonymous referees for their insightful comments and suggestions that have improved the content of this article.

© 2006 American Statistical Association
Journal of the American Statistical Association
March 2006, Vol. 101, No. 473, Theory and Methods
DOI 10.1198/016214505000000312

having 35 categorical attributes. According to scientists, these data points fall into four clusters: diaporthe stem canker, charcoal rot, rhizoctonia root rot, and phytophthora rot, with breakdowns of 10, 10, 10, and 17 data points. The reason that we chose this dataset for illustration is that the data points have relatively high dimensionality (35 attributes) in comparison to the sample size (47); this implies that these data points might be well separated from each other. The classification rates given by the K-modes algorithm are 75.72% with $K = 3$, 84.44% with $K = 4$, and 82.84% with $K = 5$. Even with the correctly specified number of clusters ($K = 4$), the algorithm still has about a 15% classification error. Compared with the method proposed in this article, which gives a 100% classification rate for the same data, the performance of the K-modes algorithm is clearly unsatisfactory. More details of the related analysis are given in Section 4.2.

Cheeseman and Stutz (1995) proposed a model-based algorithm different from the distance-based K-modes algorithm, termed *AutoClass*. The AutoClass algorithm is formulated on the basis of mixture models in that the model space constitutes all possible forms of probability density functions induced from different numbers of mixture components or modes. As a Bayesian unsupervised clustering algorithm, AutoClass can automatically determine the most probable set of partition for a given dataset through a Bayesian model selection procedure. It selects the desired model as the one whose probability density function form has the "best" posterior probability. The computational burden of this algorithm is enormous, arising mainly from two sources. First, the search for the maximum posterior parameters of the interclass models requires some computationally expensive optimization procedures, simply due to the existence of many local maxima. To determine the global maximum, AutoClass must undertake multiple searches with different starting values. Second, evaluating the posterior density of a probability density functional form requires computing a high-dimensional integration on a number of parameter sets, which is computationally intricate. To overcome such numerical difficulty, AutoClass modifies the EM algorithm in the Bayesian context and used an approximate version of the posterior density. The EM algorithm is known to have a rather slow convergence rate, which effectively keeps AutoClass from being applied to large datasets. Moreover, although Cheeseman and Stutz (1995) demonstrated through the analysis of several datasets that the approximation used in AutoClass seemed to work reasonably well in ranking candidate models, their method remains an approximation and is very computationally intensive.

We applied AutoClass to the zoo data downloaded from the same website as the soybean disease data. The zoo data consist of 101 observations, with each data point constituting 16 categorical attributes. According to specialists, the 101 animals can be classified into 7 types (mammals, birds, reptiles, fish, amphibians, insects and mollusks), with the partition of 41, 20, 5, 13, 4, 8, and 10. The sizes of clusters vary widely, ranging from 4 to 41. The sparseness in some clusters challenges the mixture model-based approach, because estimation for the parameters of the mixture proportions can be very poor in this case. Using AutoClass with multinomial distributions for the attributes, we found that this algorithm discovered only three clusters with a 73% classification rate (see Sec. 4.3 for more details). The algorithm proposed in this article correctly detects seven clusters with a 95% classification rate. Therefore, we are encouraged to present a detailed exposition of the statistical theory behind our algorithm in the rest of this article.

In this article we focus on categorical data with nominal scales and develop a clustering algorithm for such data with no use of any parametric models for attributes and any convergence criterion. We adopt the Hamming distance (HD) (Macwilliams 1978, p. 23) to form distance vectors that represent frequency distributions of data objects from a given location in the Hamming metric space. Being a measure for similarity or dissimilarity between two data objects with categorical attributes, the HD is a well-known metric in the coding theory used to infer the source that a signal or a coding sequence is originally sent. This distance is also applied in many areas, including bioinformatics, to develop clustering algorithms (e.g., Gąsieniec, Jansson, and Lingas 2004; Laboulais, Ouli, Le Bret, and Gabarro-Arpa 2002; Gabarro-Arpa and Revilla 2000). An essential difference of our algorithm from the existing methods is that it uses frequency distributions of the HD, rather than the distance itself, to determine clustering patterns. Moreover, our algorithm proceeds sequentially, and at each step it strives to detect only one cluster, which is then deleted from the current dataset in the next search. This procedure continues until there are no more significant clusters in the remaining data. To extract one cluster, we first compute HD vectors for categorical attributes, then use these to form modified chi-squared statistics with respect to each position in the categorical space. The chi-squared–type statistics are used to locate the cluster center and to determine the cluster radius. As a result, our algorithm automatically produces the number of clusters, and the resulting partition is unique.

The article is organized as follows. Section 2 presents a general framework, in which we first define HD vectors and then discuss their related properties. Section 3 introduces the clustering algorithm based on HD vectors. Section 4 illustrates the application of our algorithm in both simulated and real-world datasets, with thorough comparisons with two popular K-modes and AutoClass algorithms. Section 5 presents results of computational complexity analysis and run times for our algorithm and its two competitors. Finally, Section 6 gives some concluding remarks and a discussion. All of the technical details are deferred to the Appendix.

2. FRAMEWORK

In this section we first introduce the categorical sample space, on which we adopt the HD function to measure the relative position of two categorical data points. We then define the HD vector, which serves primarily as a distance function useful in constructing a Pearson chi-squared–type statistic in Section 3.

2.1 Categorical Sample Space

According to Anderberg (1973), two measurements with nominal scales can be distinguished only by their categories. That is, with respect to subjects 1 and 2, one can only ascertain $X_1 = X_2$ or $X_1 \neq X_2$, where X_i is the measurement of an attribute for subject i, $i = 1, 2$. For instance, the attribute may

be *gender*, which can be discernible only by the categories of male and female between two subjects.

Now consider a general setup where p nominal categorical attributes are of interest and the jth attribute is categorized by m_j levels collected by set $A_j = \{a_{j1}, \ldots, a_{jm_j}\}, j = 1, \ldots, p$. The a_{jl}'s are referred to as *states* or *modalities*. A categorical dataset, $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)$, is collected from n subjects, with $\mathbf{X}_i = (X_i^{[1]}, \ldots, X_i^{[p]})^T$ being the vector of the observed states of p attributes for subject i. Let $\mathbf{Y}_j = (X_1^{[j]}, \ldots, X_n^{[j]})^T$ be the vector of the observed states of the jth attribute from the n subjects, $j = 1, \ldots, p$. The categorical sample space, Ω_p, is defined as a collection of all possible p-dimensional vectors of states, namely $\Omega_p = A_1 \times A_2 \times \cdots \times A_p$ or

$$\Omega_p = \{(\omega_1, \ldots, \omega_p)^T | \omega_1 \in A_1, \ldots, \omega_p \in A_p\},$$

where the subscript p indicates the dimension of the space that equals to the number of attributes.

The soybean disease data were collected from $n = 47$ experimental units with $p = 35$ categorical attributes. The zoo dataset comprises $n = 101$ animals measured by $p = 16$ attributes. One attribute for the zoo data is *legs*, with the corresponding set $A_j = \{0, 2, 4, 5, 6, 8\}$, and thus this attribute has $m_j = 6$ states. Given that the other 15 attributes are binary, the sample space for the zoo data contains $6 \times 2^{15} = 196{,}608$ elements or positions. The number of positions will increase dramatically as the number of attributes increases. However, in the proposed algorithm we consider only an important subset of Ω_p, the collection of all positions that contain at least one data point and their neighboring positions. This subset is usually much smaller than the entire categorical space when the majority of the data points lie in close proximity of cluster centers. A detailed description of how to obtain this subset is given in Section 3.3.

Each element of the sample space Ω_p constitutes p components or coordinates, each from one of the A_j's. This element defines a position or a location of the sample space, representing a possible observation that happens to be located at this position.

Clearly there is no natural origin of the sample space, because the states of attributes are exchangeable and hence have no ordering. Following Gordon (1999), we adopt the indicator of mismatch as a distance function in the categorical space. Precisely, for two positions in the sample space Ω_p, $\mathbf{X}_h = (\omega_h^{[1]}, \ldots, \omega_h^{[p]})^T$ and $\mathbf{X}_i = (\omega_i^{[1]}, \ldots, \omega_i^{[p]})^T$, the HD (or metric) between $\mathbf{X}_h$ and $\mathbf{X}_i$ on the jth attribute is

$$d_j(\mathbf{X}_h, \mathbf{X}_i) = \begin{cases} 0 & \text{if } \omega_h^{[j]} = \omega_i^{[j]} \\ 1 & \text{if } \omega_h^{[j]} \neq \omega_i^{[j]}, \end{cases}$$

and the distance between the two positions is the sum of the componentwise distances,

$$d(\mathbf{X}_h, \mathbf{X}_i) = \sum_{j=1}^{p} d_j(\mathbf{X}_h, \mathbf{X}_i). \tag{1}$$

Note that the distance function in (1) is a proper metric distance function because it satisfies the following three properties:

(a) $d(\mathbf{X}_h, \mathbf{X}_i) = 0$ if and only if $\mathbf{X}_h$ and $\mathbf{X}_i$ are identical.
(b) $d(\mathbf{X}_h, \mathbf{X}_i) = d(\mathbf{X}_i, \mathbf{X}_h)$.
(c) $d(\mathbf{X}_h, \mathbf{X}_i) \leq d(\mathbf{X}_h, \mathbf{X}_l) + d(\mathbf{X}_l, \mathbf{X}_i)$ for any element $\mathbf{X}_l \in \Omega_p$.

Properties (a) and (b) are trivial, and property (c) is true because this inequality holds componentwise. With respect to component j, the inequality is always true if $\omega_h^{[j]} = \omega_i^{[j]}$ or $\omega_h^{[j]} \neq \omega_i^{[j]}$, because in the latter case it is impossible for $\omega_h^{[j]} = \omega_i^{[j]} = \omega_l^{[j]}$ to occur.

Therefore, we define a proper metric space (Ω_p, d) as the categorical sample space. Given any two data points in Ω_p, their distance d can take any integer value between 0 and p, say q. This means that the two data points differ by q attributes or, equivalently, that they coincide by $p - q$ attributes. Such a distance appropriately reflects their relative positions in the sample space.

2.2 Hamming Distance Vector

Any given dataset $\mathbf{X}$ will provide a distribution of distances in the sample space Ω_p with respect to a fixed reference position in this space. We now introduce the HD vector to summarize such information for a dataset.

Definition 1. Let $\mathbf{S} = (s_1, s_2, \ldots, s_p)$ be a reference position in the space (Ω_p, d). The HD vector is a $(p+1)$-element vector $\mathbf{U}(\mathbf{S}) = [U_0(\mathbf{S}), U_1(\mathbf{S}), \ldots, U_p(\mathbf{S})]^T$ where the qth component is the frequency given by

$$U_q(\mathbf{S}) = \sum_{j=1}^{n} I[d(\mathbf{X}_j, \mathbf{S}) = q], \qquad q = 0, 1, \ldots, p. \tag{2}$$

As usual, here $I[E]$ is the indicator function that equals 1 if event E occurs.

Note that $U_q(\mathbf{S})$ counts the number of all data points in $\mathbf{X}$ that appear to have the same distance q to the given reference position $\mathbf{S}$, or the number of all subjects that appear to have exactly q different attributes from the reference position $\mathbf{S}$. In particular, $U_0(\mathbf{S})$ is the number of data points in $\mathbf{X}$ that are identical to the reference $\mathbf{S}$. Clearly $\sum_{q=0}^{p} U_q(\mathbf{S}) = n$ or $\sum_{q=0}^{p} \{U_q(\mathbf{S})/n\} = 1$, irrespective of the reference $\mathbf{S}$.

In principle, the reference point $\mathbf{S}$ is arbitrary, because Ω_p has no origin. Once the reference $\mathbf{S}$ is fixed, the vector $\mathbf{U}(\mathbf{S})$ gives

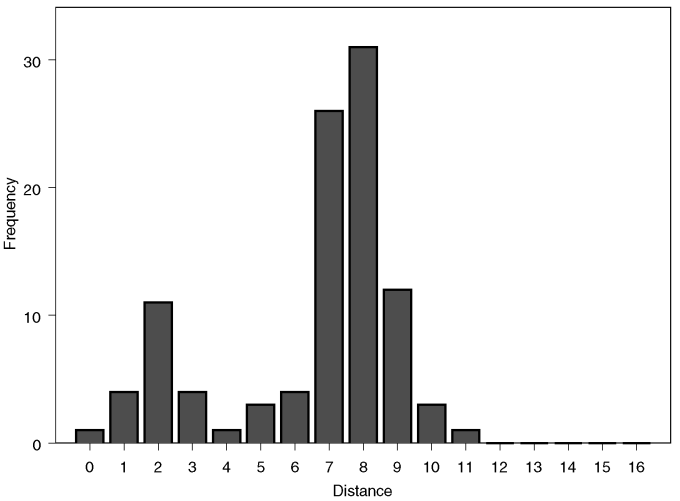

Figure 1. One HD Vector for the Zoo Dataset.

a frequency distribution of the distances with the support of integers from 0 to p. Figure 1 shows a distribution of this type for the zoo data that we use in the analysis given in Section 4.3. It is interesting to observe that this distribution appears to be bimodal, suggesting the existence of certain clustering patterns in the zoo data. Moreover, with a given distribution of relative frequencies, $U_q(\mathbf{S})/n$, $q = 0, 1, \ldots, p$, we can calculate the expected distance to the reference position $\mathbf{S}$ as

$$\mu_d(\mathbf{S}) = \sum_{q=0}^{p} q \frac{U_q(\mathbf{S})}{n}.$$

A key step in extracting a cluster is to determine the cluster center. Following the idea of modes of Huang (1997), we define a data center as follows.

Definition 2. A position $\mathbf{C} = (c_1, c_2, \ldots, c_p) \in \Omega_p$ is said to be the center of a dataset $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)$ if it minimizes the sum of the HDs over all data points, namely

$$D(\mathbf{C}, \mathbf{X}) = \sum_{i=1}^{n} d(\mathbf{C}, \mathbf{X}_i) = \min_{\mathbf{S} \in \Omega_p} \sum_{i=1}^{n} d(\mathbf{S}, \mathbf{X}_i). \tag{3}$$

The following theorem demonstrates that such an HD function-based definition of the data center has a frequency interpretation for a categorical dataset.

Theorem 1. Suppose that $\mathbf{C} = (c_1, c_2, \ldots, c_p) \in \Omega_p$ is the center of a categorical dataset $\mathbf{X}$. Then the jth component c_j is one state in A_j that occurs most frequently in the vector $\mathbf{Y}_j = (X_1^{[j]}, \ldots, X_n^{[j]})^T$ for the jth attribute, $j = 1, 2, \ldots, p$.

In general, the center for a categorical dataset might not be unique because of multiple modes.

Theorem 2. A categorical data center $\mathbf{C}$ gives the minimal expected distance, namely

$$\mu_d(\mathbf{C}) = \min_{\mathbf{S} \in \Omega_p} \mu_d(\mathbf{S}).$$

Theorem 2 implies that the frequency $U_q(\mathbf{C})$ with respect to the data center $\mathbf{C}$ tends to be large with a small distance q and tends to be small with a large distance q, because $\sum_{q=0}^{p} q U_q(\mathbf{S})$ must reach the minimum at the position $\mathbf{C}$. For instance, an HD vector for the zoo data shown in Figure 1 displays virtually zero frequencies at the five largest distances, 12–16.

2.3 Uniform HD Vector

To use the HD vector to detect possible clusters, we first need to define a reference HD vector that reflects the condition of no clustering pattern in the categorical sample space Ω_p. From a frequency standpoint, we refer to a situation in which all data points have equal probability of occurring at each position in the space as the case of no clustering pattern. The resulting HD vector is called the *uniform* HD vector (UHD) (see Definition 3), which represents the distribution of the expected frequencies under the null hypothesis that there are no clustering patterns in the data. Unlike model-based clustering algorithms, in this article we do not assume any parametric distributions for data, and in such a setting the empirical distribution (i.e., the frequency distribution) appears to be appealing for establishing the procedure to test for the existence of clustering patterns. As known in classical statistical theory, the empirical distribution, in the form of, say, a histogram, has proven to be a powerful tool for describing the underlying data-generation mechanism.

Although there might be some alternative reference (or null) distributions to be considered, from a probability standpoint, the UHD seems suitable and feasible for the development of our algorithm.

Definition 3. Let $\mathbf{X}$ be a categorical dataset of n random observations, with each observation having an equal probability of locating at any position on space Ω_p. The expected value of the HD vector associated with such data, denoted by $E\{\mathbf{U}(\mathbf{S})\}$, is referred to as the UHD vector.

Clearly the total number of possible positions in a categorical sample space is $M = \prod_{j=1}^{p} m_j$, where $m_j, j = 1, 2, \ldots, p$, is the number of states in set A_j for the jth attribute. The following theorem gives an expression of the UHD vector.

Theorem 3. The UHD vector is given by $\boldsymbol{\mathcal{E}} = (\frac{n}{M})\mathbf{U}^*$, where $\mathbf{U}^* = (U_0^*, U_1^*, \ldots, U_p^*)^T$ with

$$\begin{aligned}
U_0^* &= 1, \\
U_1^* &= (m_1 - 1) + (m_2 - 1) + \cdots + (m_p - 1), \\
U_2^* &= \sum_{i<j}^{p} (m_i - 1)(m_j - 1), \\
&\vdots \\
U_p^* &= (m_1 - 1)(m_2 - 1) \cdots (m_p - 1).
\end{aligned}$$

Moreover, the UHD vector is independent of position $\mathbf{S} \in \Omega_p$.

Figure 2 shows the UHD for the zoo data, which represents the distribution of the expected frequencies at distances 0–16. This distribution appears to be approximately bell-shaped and symmetric around the median distance $(p + 1)/2$, due to the pattern of the constants, U_j^*'s. In fact, this property holds in general.

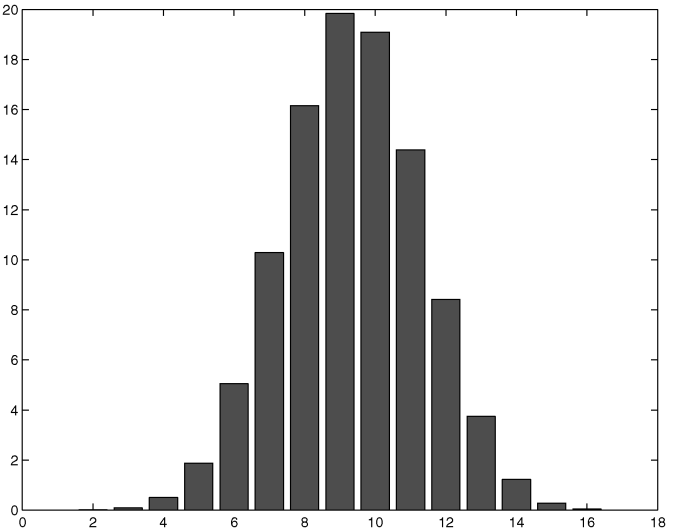

Figure 2. The UHD Vector for the Zoo Dataset.

3. ALGORITHM

We have two tasks to accomplish in our algorithm: to examine whether there are any clustering patterns in the data, and to extract the clusters if they indeed exist in the data. For the first task, we propose testing for the null hypothesis H_0: There is no clustering pattern in the data. Under the H_0 (namely, the data being randomly distributed in space Ω_p), the UHD vector represents the distribution of the distance frequencies. Therefore, a large difference between the HD vector $\mathbf{U}(\mathbf{S})$ of the observed data and the UHD vector $\boldsymbol{\mathcal{E}}$ would indicate statistical evidence against the null hypothesis H_0. Hence rejecting the H_0 leads us to conclude the existence of some clustering structures in the data. We adopt Pearson's chi-squared test for this testing problem.

3.1 Testing for Clustering Patterns

Pearson's chi-squared statistic is commonly used to measure the discrepancy between two frequency distributions defined on a finite discrete sample space. Let $\boldsymbol{\mathcal{E}} = (\mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_p)^T$ be the UHD vector and let $\mathbf{U}(\mathbf{S}) = (U_0(\mathbf{S}), U_1(\mathbf{S}), \ldots, U_p(\mathbf{S}))^T$ be the HD vector with respect to position $\mathbf{S}$. Pearson's chi-squared statistic takes the form of

$$\chi^2(\mathbf{S}) = \sum_{j=0}^{p} \frac{(U_j(\mathbf{S}) - \mathcal{E}_j)^2}{\mathcal{E}_j}, \qquad \mathbf{S} \in \Omega_p,$$

which follows approximately a $\chi^2(p)$ distribution. Note that this chi-squared statistic is available for every position $\mathbf{S} \in \Omega_p$. But at most positions in Ω_p, the statistic is not very informative unless it is evaluated at data centers. This is because the HD vector $\mathbf{U}(\mathbf{S})$ evaluated at a data center $\mathbf{C}$ will produce the maximal contrast to the UHD vector, as desired. Therefore, this testing for clustering patterns becomes the problem of finding data centers.

Intuitively, if the dataset contains clusters, then a cluster center is supposedly a position around which there is a certain data cloud. When one starts at a cluster center and travels outward, one would pass through layers of denseness of data points in the space. Frequencies actually reflect the spectrum of the denseness along distances from the data center. Moreover, the frequencies at low distance values will appear to have a discernible local "bump"; see Figure 1, which contains a local bump around distance 2.

Conversely, if there are no clustering patterns in the space, then, departing from the same position, one would see (nearly) uniformly distributed data points in the space. As a result, the frequencies at low distance values will not appear to have any local bumps, but the only global peak will occur at the median distance (see Fig. 2).

The next theorem claims that as long as the data points are not uniformly distributed in the sample space, a local bump pattern, which reflects higher frequencies than the UHD at small distance values, exists for at least one distance value.

Theorem 4. Assume that the data $\mathbf{X}$ are not uniformly distributed in the sample space Ω_p. Let $\{\mathbf{U}(\mathbf{S}) = (U_0(\mathbf{S}), U_1(\mathbf{S}), \ldots, U_p(\mathbf{S}))^T, \mathbf{S} \in \Omega_p\}$ be the collection of all $(p+1)$-element HD vectors in the space Ω_p, and let $\boldsymbol{\mathcal{E}} = (\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_p)^T$ be the UHD vector given in Theorem 3. Then for a given distance value q, $q = 0, 1, \ldots, p$, there always exists at least one position, $\mathbf{S}_q^* \in \Omega_p$, such that the frequency at this distance value is larger than the corresponding component, ε_q, of the UHD vector $\boldsymbol{\mathcal{E}}$, namely

$$U_q(\mathbf{S}_q^*) > \varepsilon_q \quad \text{for some } \mathbf{S}_q^* \in \Omega_p.$$

This theorem implies that, reflected on the HD vector, the $\mathbf{U}(\mathbf{C})$ with respect to a data center would be likely to have large frequencies at small distance values and small frequencies at large distance values. Compared with the UHD vector, the discrepancy will be overwhelming for the first few terms between the observed HD vector and the UHD vector, and the contribution from the rest of the terms will be more relevant for the other clusters that are "far away" from the cluster under investigation.

To proceed, we need to decide how many and which low distance values should be included in the comparison. Let r be a cutoff point such that the "bump" pattern resides on the interval $(0, r)$. Choosing an optimal value of r in the partition of the chi-squared statistic is generally difficult. This is because it is hard to precisely quantify the cutoff point between the relevant and irrelevant distance frequencies in relation to an underlying clustering structure. Instead, we introduce a selection criterion based on Theorem 2, which proves that in the presence of clusters, the early segment of an HD vector with respect to a data center should contain substantially larger frequencies than the corresponding frequencies of the UHD vector. Thus the range on which the HD vector is consistently larger than the UHD vector gives a reasonable indication of the r. This leads to

$$r_t^*(\mathbf{S}) = \min_{j>0} \left\{ j \,\middle|\, \frac{U_j(\mathbf{S})}{\mathcal{E}_j} < 1 \right\} - t, \qquad \mathbf{S} \in \Omega_p, \tag{4}$$

where t is a tuning constant, $t = 1, \ldots, T$, for a suitable T, which is useful for tuning the upper cutting edge. For instance, when $t = 1$, $r_1^*(\mathbf{S})$ indicates the upper limit of the range on which the HD vector is larger than the UHD vector. Precisely, at the distance value $r = r_1^*(\mathbf{S})$, $U_{r_1^*(\mathbf{S})}(\mathbf{S}) \geq \mathcal{E}_{r_1^*(\mathbf{S})}$, but at the immediately next distance value, $r = r_1^*(\mathbf{S}) + 1$, $U_{r_1^*(\mathbf{S})+1}(\mathbf{S}) < \mathcal{E}_{r_1^*(\mathbf{S})+1}$. In fact, this tuning constant can be automatically determined by the criterion of the most significant contrast between the observed HD vector and the UHD vector. That is, with a given position $\mathbf{S}$, the optimal cutoff point $r^*(\mathbf{S})$ is the one that gives the smallest p value of the modified chi-squared statistic defined in Theorem 5, namely

$$r^*(\mathbf{S}) = \arg\min_{r_t^*} \{p\text{-value of } \chi_M^2(r_t^*; \mathbf{S})\},$$

where $\chi_M^2(r_t^*; \mathbf{S})$ is given in (5). In the case where there does not exist a local bump at low distance values, we assign $r^*(\mathbf{S}) = 0$.

Figure 3 uses the HD and UHD vectors of Figures 1 and 2 to give a graphical demonstration for the choice of a cutoff value r^*, indicated by the solid vertical line in the zoo data analysis.

We now consider partitioning the Pearson chi-squared statistic into two parts with a cutoff point r as follows:

$$\chi^2(\mathbf{S}) = \sum_{j=0}^{r} \frac{(U_j(\mathbf{S}) - \mathcal{E}_j)^2}{\mathcal{E}_j} + \sum_{j=r+1}^{p} \frac{(U_j(\mathbf{S}) - \mathcal{E}_j)^2}{\mathcal{E}_j}.$$

For the ease of exposition, we apply the optimal $r^*(\mathbf{S})$ for r at the end of this discussion.

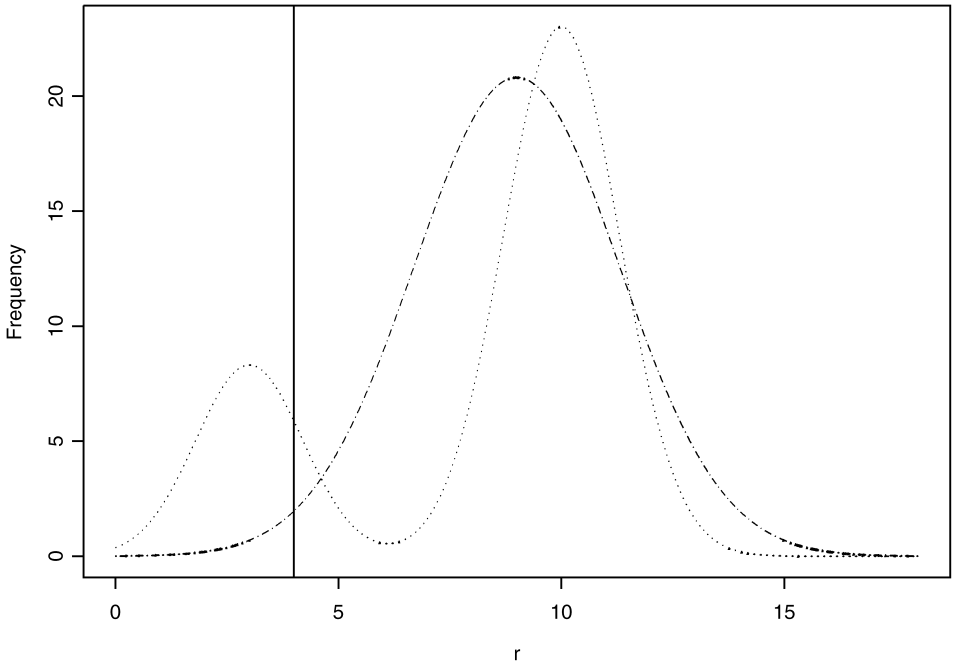

Figure 3. Choosing the Cutoff Value r^* in the Zoo Data Analysis, Indicated by the Solid Vertical Line (—, r^*; · · ·, HD; - · -, UHD).

The first part of the $(r+1)$ terms contains the most relevant information for finding a data center; the remaining part is not relevant to the local bump pattern, and including it will greatly reduce the testing power. Therefore, this second part should be neutralized such that it is included only for the valid allocation of the data. As a result, we propose the following modified chi-squared statistic:

$$\chi^2_M(r;\mathbf{S}) = \sum_{j=0}^{r} \frac{(U_j(\mathbf{S}) - \mathcal{E}_j)^2}{\mathcal{E}_j} + \min_{\{U_j(\mathbf{S}),\, r<j\le p\}} \sum_{j=r+1}^{p} \frac{(U_j(\mathbf{S}) - \mathcal{E}_j)^2}{\mathcal{E}_j}, \quad (5)$$

with the constraint that $\sum_{j=r+1}^{p} U_j(\mathbf{S}) = n - \sum_{j=0}^{r} U_j(\mathbf{S})$.

Clearly, the modified chi-squared statistic gives the least significant value when the first $(r+1)$ terms, $U_0(\mathbf{S}), U_1(\mathbf{S}), \ldots, U_r(\mathbf{S})$, are given. For any given r, the following theorem shows that the modified $\chi^2_M(r;\mathbf{S})$ statistic effectively merges the last $p-r$ terms into one term. Consequently, a traditional chi-squared test can be performed on the basis of a subset of the HD vector.

Theorem 5. With a given r, the modified chi-squared statistic takes the form

$$\chi^2_M(r;\mathbf{S}) = \sum_{j=0}^{r} \frac{(U_j(\mathbf{S}) - \mathcal{E}_j)^2}{\mathcal{E}_j} + \frac{(\sum_{j=0}^{r} U_j(\mathbf{S}) - \sum_{j=0}^{r} \mathcal{E}_j)^2}{\sum_{j=r+1}^{p} \mathcal{E}_j}. \quad (6)$$

Because the modified chi-squared statistic is constructed to capture a local clustering pattern, rather than assessing the overall goodness of fit, it is useful to determine a cluster center. That is, with a given r, we choose the position that gives the maximum modified chi-squared statistic as the data center.

When the optimal cutoff value $r^*(\mathbf{S})$ is applied, the resulting modified chi-squared statistic in (6) based on $r^*(\mathbf{S})$ is given by

$$\chi^{2*}_M(\mathbf{S}) = \chi^2_M(r^*(\mathbf{S});\mathbf{S}), \qquad \mathbf{S} \in \Omega_p. \quad (7)$$

Consequently, a cluster center $\mathbf{C}$ is naturally chosen as the position at which $\chi^{2*}_M(\mathbf{S})$ attains the maximum, that is,

$$\mathbf{C} = \arg\max_{\mathbf{S}\in\Omega_p} \chi^{2*}_M(\mathbf{S}).$$

3.2 Estimating Cluster Radius

When statistical evidence about the existence of clustering patterns in a dataset is found, extracting clusters is the next task. To completely determine a cluster, we need to specify its center and size. In the previous section we developed a procedure for detecting a cluster center, and in this section we introduce a method for determining the physical size of a cluster. To do so, we first define the radius of a cluster on space (Ω_p, d).

Definition 4. Let Φ denote a cluster with center $\mathbf{C}$. The radius R of the cluster Φ is the maximum distance of the data points in this cluster to its center, that is,

$$R = \max_{\mathbf{X}_i\in\Phi} d(\mathbf{X}_i, \mathbf{C}).$$

Estimating the radius of a cluster has long been a challenge in clustering analysis, because cluster boundaries can be very complicated for some datasets. To proceed, certain assumptions on the cluster structures are inevitable. In this article we consider a scenario that we learned from the analyses of the soybean disease data and the zoo data; that is, when a cluster center is given, the density of the data points around the center position first increases, then decreases in this cluster moving outward from the center. The distance frequencies in the HD vector relevant to this cluster appear to have a local parabolic curvature. Figure 1 gives one example of this situation. Moreover, if the

increase and decrease rates are approximately equal, then the local curvature will be approximately symmetric around a local peak that corresponds to the thickest layer of the data points in this cluster. In this situation, it seems reasonable to estimate the radius R as the distance at which the HD vector has its very first local minimum, that is,

$$R(\mathbf{C}) = \min_{0<j<p} \{j | U_j(\mathbf{C}) < \min(U_{j-1}(\mathbf{C}), U_{j+1}(\mathbf{C}))\} - 1. \quad (8)$$

The radius, $R(\mathbf{C})$, is usually smaller than the partition cutoff, $r^*(\mathbf{C})$, in the modified chi-squared statistic. The difference between $R(\mathbf{C})$ and $r^*(\mathbf{C})$ depends on the curvature shape of the frequencies of the HD vector at distance values $0, \ldots, r^*(\mathbf{C})$. The more skewed the local curvature, the higher the misclassification rate of the algorithm. In effect, when the local curvature has a long right tail, $R(\mathbf{C})$ may be overestimated; conversely, when the local curvature has a long left tail, $R(\mathbf{C})$ may be underestimated. Either case would increase the misclassification rate.

3.3 Termed as HD Vector Algorithm

The proposed algorithm, the HD vector algorithm, proceeds as follows:

Step 1. Determine $\mathcal{D}_p$, the set of distinct positions in the sample space Ω_p, and compute the corresponding frequency at each of such positions in $\mathcal{D}_p$. The set $\mathcal{D}_p$ consists of $\mathcal{O}_p$, the set of positions occupied by the observed data points, and $\bigcup_{\mathbf{S}\in\mathcal{O}_p} \mathcal{N}_e(\mathbf{S})$, the collection of the positions that are e-distance neighbors to each of the positions in $\mathcal{O}_p$, namely

$$\mathcal{N}_e(\mathbf{S}) = \{\mathbf{S}' | d(\mathbf{S}', \mathbf{S}) \le e\}, \qquad \mathbf{S} \in \mathcal{O}_p.$$

In both our simulation and data analyses, $e = 1$ was chosen.

Step 2. For each position $\mathbf{S} \in \mathcal{D}_p$, calculate the HD vector $\mathbf{U}(\mathbf{S})$, determine the cutoff $r^*(\mathbf{S})$, and obtain the corresponding modified chi-squared statistic $\chi^{2*}_M(\mathbf{S})$. If $r^*(\mathbf{S}) = 0$, then assign $\chi^{2*}_M(\mathbf{S}) = 0$ and label this $\mathbf{S}$ as an isolated position.

Step 3. Select and compare the largest statistic, $\max_{\mathbf{S}\in\mathcal{D}_p} \chi^{2*}_M(\mathbf{S})$ to the critical value of $\chi^2_{[.05]}(r^*(\mathbf{S}) + 1)$. If the maximum is smaller than the critical value, then stop the algorithm; otherwise, continue with the next step.

Step 4. Determine the position corresponding to the maximum modified statistic as a cluster center

$$\mathbf{C} = \arg\max_{\mathbf{S}\in\mathcal{D}_p} \chi^{2*}_M(\mathbf{S}),$$

calculate the corresponding radius by (8), label all data points in the cluster, and remove them from the current dataset.

Step 5. Repeat Steps 2–4 until either there is no more significant cluster center or there are only isolated data points remaining.

It is worth pointing out that this proposed algorithm automatically outputs the number of clusters and does not involve any judgment of convergence. Therefore, it avoids dealing with the problem of multiple local maxima, which is very computationally expensive in the model-based algorithms such as AutoClass, which uses the EM algorithm. Moreover, compared with the popular K-modes algorithm, our algorithm is insensitive to the order of data input and selects the most significant cluster at each iteration, and thus it produces a unique partition of the data. Moreover, our algorithm is numerically efficient and highly accurate, and it outperforms these two popular algorithms for clustering categorical data, as shown in the numerical experiments based on both simulated and real-world datasets in the next section.

4. NUMERICAL EXPERIMENTS

In this section we illustrate our algorithm on both simulated and real-world datasets. In particular, we compare the proposed HD vector algorithm with two currently popular algorithms, AutoClass and K-modes, with both classification rate and information gain used to evaluate their performance.

A natural standard for assessing the quality of performance is the classification rate that essentially measures the accuracy of an algorithm to assign data points into correct clusters. With given K clusters, the classification rate (CR) is defined by

$$CR(K) = \sum_{k=1}^{K} \frac{\widetilde{n}_k}{n},$$

where n is total number of data points and $\widetilde{n}_k$ is the number of data points that have been correctly assigned to cluster k by an algorithm. Obviously, $0 \le CR(K) \le 1$, and a larger $CR(K)$ value indicates better performance of clustering.

An alternative criterion for assessing the performance of a clustering algorithm is the so-called *cluster purity* proposed by Bradley, Fayyad, and Reina (1998). Cluster purity essentially measures the information gain (IG), which is the difference between the total entropy and weighted entropy for a given data partition, namely

$$\text{information gain}(IG(K)) = \text{total entropy} - \text{weighted entropy}(K),$$

where the weighted entropy is calculated by

$$\text{weighted entropy}(K) = \sum_{k=1}^{K} \frac{n_k}{n} \times \text{cluster entropy}(k),$$

with

$$\text{cluster entropy}(k) = -\sum_{l=1}^{L} \frac{\widetilde{n}^k_l}{n_k} \log_2 \left\{ \frac{\widetilde{n}^k_l}{n_k} \right\},$$

where $\widetilde{n}^k_l$ is the number of data points with true label l in cluster k, n_k is the number of data points known in cluster k, and L is the known number of classes. In this article we take a ratio of $IG(K)$/total entropy, which is similar to the classification rate is between 0 and 1. It is worth noting that the IG criterion may lead to a misleading conclusion. One scenario is that one algorithm perfectly splits each of the true clusters into two clusters, which is apparently an incorrect classification, but the IG is still equal to 1, indicating a perfect clustering. Such a case occurred in our simulation study, although it was rare.

Because the K-modes algorithm is known to be sensitive to the order of the data input, in contrast to the unique result given by the AutoClass and the HD vector algorithms, a sensible way to compare the K-modes algorithm with the other two algorithms must be undertaken with the average $CR(K)$ or the average $IG(K)$ over a certain number of permutations of the data

input orders. This is because averaging will reduce the dependence on the order of data input. Letting G be the number of permutations on the data input order of the same dataset, we can calculate the average classification rate (ACR), $ACR(K)$, and the average information gain (AIG), $AIG(K)$, as well as the permutation sensitivity (PS) index given by

$$PS(CR) = \sqrt{\frac{1}{G-1}\sum_{g=1}^{G}\big(CR_g(K) - ACR(K)\big)^2} \qquad \text{and}$$

$$PS(IG) = \sqrt{\frac{1}{G-1}\sum_{g=1}^{G}\big(IG_g(K) - AIG(K)\big)^2},$$

where $CR_g(K)$ and $IG_g(K)$ are the CR and the IG for the gth permutation, $g = 1, \ldots, G$. Note that the PS measures the variation due to the reordering of the data input, and it does not measure the variation caused by any underlying stochastic mechanism.

In both simulation study and data analyses, the HD vector and AutoClass algorithms allow us to estimate the number of the clusters, but the K-modes algorithm needs a prefixed number of clusters. Also, we randomly reorder the original data a number of times and report the ACR and AIG, as well as the corresponding sensitivity indices.

4.1 Simulation Experiment

The advantage of using a simulation study to validate the proposed algorithm lies in the fact that the cluster membership is fully known, so that both classification rate and information gain can be accurately calculated and used to compare the performance among the three algorithms. For simplicity, we consider only the case where all attributes are uncorrelated. We simulate data according to the following steps:

1. Set $p = 10$. Randomly select a number from the set $\{4, 5, 6\}$ as the m_j, which is the number of states for the jth categorical attribute, $j = 1, \ldots, 10$.
2. Set $K = 5$ clusters. Choose five cluster centers, $\mathbf{C}_k, k = 1, \ldots, 5$, such that $d(\mathbf{C}_k, \mathbf{C}_{k'}) \geq 5$ for all $k \neq k'$.
3. Set the sample size $n = 200$ with cluster sizes $n_1 = 70$, $n_2 = 50$, $n_3 = 40$, $n_4 = 25$, and $n_5 = 15$. In the kth cluster with center $\mathbf{C}_k = (c_{k,1}, \ldots, c_{k,10})$, generate n_k 10-attribute vectors independently. In particular, generate the jth attribute, which has m_j states in $A_j = \{a_{j1}, a_{j2}, \ldots, a_{jm_j}\}$, by a multinomial distribution with probabilities $p^j_{k,1}, \ldots, p^j_{k,m_j}$ such that the probability corresponding to the center state, $c_{k,j} \in A_j$, is .7 and the rest probabilities are identically equal to $.3/(m_j - 1)$.

Table 1 gives a summary of the simulation study with 100 replications. In each replication, the K-modes algorithm was applied to $G = 80$ reordered datasets to yield the ACR and the AIG. Note that the reordering has no effect on the results of the HD vector and AutoClass algorithms. In addition, the number of clusters was fixed as the true value $K = 5$ in the application of the K-modes algorithm.

Clearly, the HD vector algorithm outperformed the other two algorithms; it is 10% more accurate than AutoClass and 5% more accurate than K-modes, based on the ACR of the 100 simulations. A similar conclusion can be drawn using the criterion of the AIG. The estimated density functions of the ACR and the AIG based on the 100 simulations are plotted in Figure 4. The figure clearly suggests that in light of the ACR, the performance of the HD vector is the best and that of the K-modes is better than that of the AutoClass, and that in light of the AIG, the performance of the HD vector is again the best, and the other two seem close to each other. In conclusion, the HD vector algorithm looks much more appealing than the other two algorithms.

Table 1. Sample Means and Standard Deviations (SDs) of the Average Classification Rates and Average Information Gains for the HD Vector, AutoClass, and K-Modes Algorithms Over 100 Simulations

	HD vector	*AutoClass*	*K-modes*
Mean of ACR	94.62%	83.32%	90.86%
SD of ACR	3.14%	7.80%	2.14%
Mean of AIG	86.24%	82.99%	82.07%
SD of AIG	6.76%	4.54%	4.53%

It is worth pointing out that the sample standard deviation given by the K-modes algorithm looks smaller than those given by the other two algorithms. This is because in the case of the K-modes algorithm double averages were involved, because the ACR or the AIG was an average of 80 values of the CR or the IG in the first place. In fact, the K-modes algorithm gives much more variable clustering results than the other two algorithms, which can be seen from the data analyses discussed in the following two sections.

On average, the K-modes algorithm seems to perform slightly better than the AutoClass algorithm. This may be due to the nature of the data generated in this study; the five clusters were separated with at least five units of the HD d in the space, and probability .7 was assigned to generate the state corresponding to the cluster center. But in practice, some datasets may be more complex than the setup considered in our simulation study. As in the example of the soybean disease data of Section 4.2, the AutoClass algorithm turns out to be much more favorable than the K-modes algorithm.

4.2 Analysis of Soybean Disease Data

We now present a detailed analysis of the soybean disease data discussed briefly in Section 1. Because all 35 attributes of the data are categorical, this dataset is suitable for examining our algorithm with comparisons to the other algorithms, such as the AutoClass and K-modes algorithms. The original dataset contains 388 observations, but we used only 47 observations with completely observed attributes in our study. In addition, the data file lists the membership of the data points in four clusters: diaporthe stem canker, charcoal rot, rhizoctonia root rot, and phytophthora rot. The cluster sizes are 10, 10, 10, and 17.

To apply the AutoClass algorithm, we specified the multinomial distribution as the marginal distribution for each attribute. The results are summarized in Table 2. Because the K-modes algorithm is dependent on the order of data input, 100 randomly reordered datasets were generated from the original data and then analyzed by the algorithm. Here the ACR is the mean of the 100 classification rates obtained from the 100 permuted

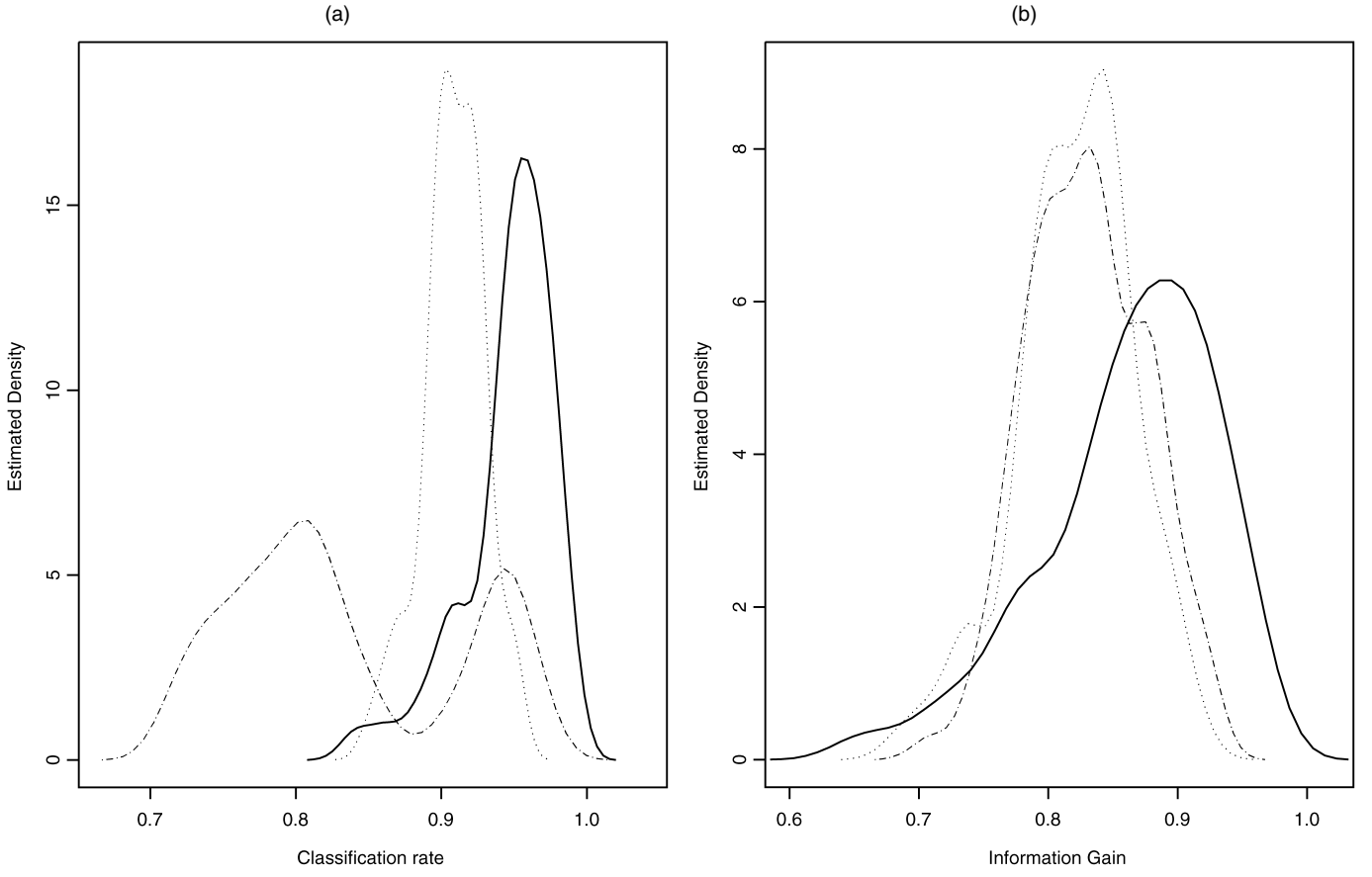

Figure 4. Estimated Density Functions of (a) the ACR and (b) the AIG Over 100 Simulations for the AutoClass (-·-·-), K-Modes (· · ·), and HD Vector (—) Algorithms.

datasets, and so it is the AIG. As seen, both HD vector and AutoClass algorithms gave the perfect clustering for the soybean data; they not only correctly estimated the number clusters, but also produced a perfect partition of the data. In contrast, the K-modes algorithm was able to assign only about 85% of the data points into the correct clusters, even when the number of clusters was chosen as the true value $K = 4$. It is worth noting once again that the data used in our analysis has a relatively small sample size, but the dimensionality of each data point is relatively high. As a matter of fact, these data points are well separated in the sample space, and thus it is not surprising that the perfect classification can be yielded by both HD vector and AutoClass algorithms.

The performance of the K-modes algorithm varies dramatically with the AIG ranging from 40% to 95%, as shown in Figure 5. Even when the true $K = 4$ was used, there was a sizable proportion of cases with rather low information gain, and on average the algorithm had only a 84.44% classification accuracy and a 15.65% permutation sensitivity.

In conclusion, both the HD vector and AutoClass algorithms substantially outperformed the K-modes algorithm, at least in the case of the soybean disease data. Moreover, the AutoClass algorithm depends on the multinomial distribution assumption for marginal attributes and some expensive numerical optimization procedures, such as the EM algorithm. Therefore, as far as the computational efficiency concerns, the HD vector algorithm is preferable to the AutoClass algorithm in the context of categorical data cluster analysis.

4.3 Analysis of Zoo Data

The zoo data present more challenges in the cluster analysis than the soybean disease data, because of a larger sample size and more variable cluster sizes. In fact, the zoo data were collected from 101 animals, each measured by 16 categorical

Table 2. Clustering Results Given by the HD Vector, AutoClass, and K-Modes Algorithms for Analysis of the Soybean Disease Data

	HD vector	*AutoClass*	*K-modes*		
	4 clusters	*4 clusters*	*[3] clusters*	*[4] (true) clusters*	*[5] clusters*
ACR	100%	100%	75.72%	84.44%	82.84%
PS(*CR*)	0%	0%	6.65%	15.65%	9.55%
AIG	100%	100%	66.81%	84.39%	93.10%
PS(*IG*)	0%	0%	10.15%	15.37%	11.30%

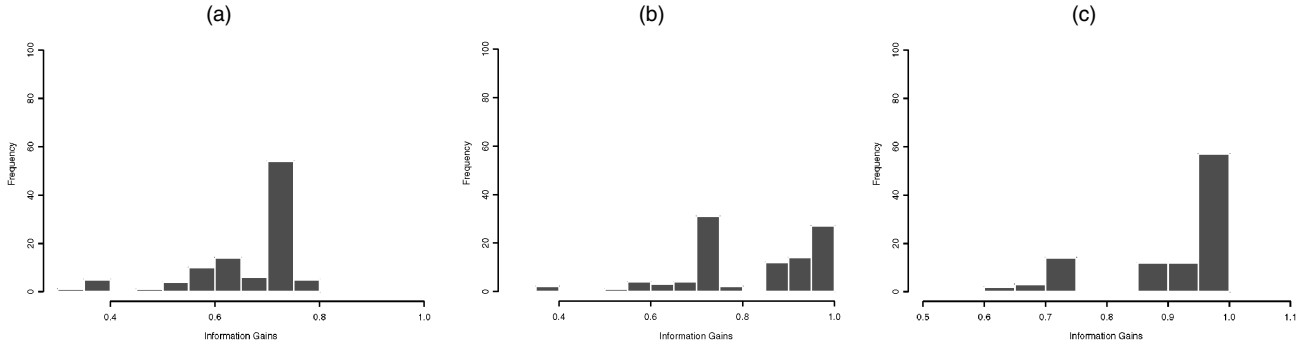

Figure 5. The Distributions of the IG Produced by the K-Modes Algorithm for the Soybean Data. The three prespecified numbers of clusters are (a) 3, (b) 4, and (c) 5.

attributes: hair, feathers, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, tail, domestic, cat-size, and legs. The first 15 attributes are dichotomous, and the legs variable is polytomous. Similar to the soybean disease data, these 101 animals were classified by specialists into 7 types (mammals, birds, reptiles, fish, amphibians, insects, and *mollusks*), with cluster sizes $41, 20, 5, 13, 4, 8$, and 10. Note that there are two clusters that contain only four and five observations, and such sparseness usually challenges model-based clustering methods. For example, it is hard for the AutoClass algorithm to obtain good estimation for the parameters of mixture proportions in the mixture models.

We used the HD vector, AutoClass, and K-modes algorithms to analyze these data; the results are reported in Table 3. Again, here we let the number of clusters vary in the application of the K-modes algorithm around the true value $K = 7$. In addition, we randomly shuffled the order of data input 100 times for the K-modes method, to remove the dependence of this method on the choice of the data input order.

Remarkably, the HD vector correctly identified seven clusters, whereas the AutoClass algorithm detected only three clusters. This is because several clusters are too sparse to be discerned by the AutoClass algorithm. In the comparison of classification accuracy, the HD vector algorithm was capable of correctly assigning 96 out of 101 observations into their own clusters, which is about 20% more accurate than the other two algorithms. In this study, both the AutoClass and K-modes algorithms performed poorly, with only about 73% classification accuracy.

Our conclusions from this analysis are as follows: 1. The HD vector algorithm is highly accurate and outperforms the other two algorithms, with both CR rate and IG improved by several orders of magnitudes. 2. The K-modes algorithm is slightly better than the AutoClass algorithm on the basis of either the AIG or the ACR; however, the K-modes algorithm apparently lacks robustness against the data input order.

5. COMPUTATIONAL COMPLEXITY AND RUN TIMES

In this section, we discuss the computational complexity of the AutoClass, K-modes, and HD vector algorithms. As before, we denote the sample size by n, the number of attributes by p, and the number of clusters by K.

The AutoClass algorithm is the most flexible of these three algorithms; it can handle both quantitative and categorical data types, as well as missing data. However, it uses the EM algorithm as the "engine" for parameter estimation in that maximizing the likelihood function is carried out iteratively. The computational complexity of the AutoClass algorithm is approximately $O(n\ p\ K\ l\ a)$, where l is the average number of iterations required for the EM algorithm and a is the average number of iterations needed for the optimization step in the EM algorithm. It is well known that convergence can be rather slow in the EM algorithm; therefore, $l \times a$ may be quite large in practice.

Because the K-modes algorithm is considered as a variant of the K-means algorithm, the computational complexity for both algorithms should be the same. It is known that the computational complexity for the K-means algorithm is $O(n\ p\ K\ l)$, where l is the number of iterations required for convergence. This term l could make the K-means algorithm rather slow, as pointed out by many authors in the data-mining literature. For example, Bradley et al. (1998) pointed out that the K-means algorithm does not scale well to large datasets.

The proposed HD vector algorithm has the computational complexity of $O(n\ p\ K\ e)$, where e is the size of the neighboring positions used in the clustering. In both simulation and data analyses, we chose $e = 1$, which enabled us to achieve very high

Table 3. Clustering Results Given by the HD Vector, AutoClass, and K-Modes Algorithms for Analysis of the Zoo Data

	HD vector	*AutoClass*	*K-modes*		
	7 clusters	*3 clusters*	*[6] clusters*	*[7] (true) clusters*	*[8] clusters*
ACR	95.05%	73.27%	74.40%	72.27%	70.8%
PS(CR)	0%	0%	10.13%	9.78%	9.40%
AIG	91.59%	60.38%	75.90%	79.11%	81.21%
PS(IG)	0%	0%	6.86%	5.96%	6.37%

Table 4. Summary of the Run Times (in seconds) for the Soybean Data Analysis With $n = 47$ and $p = 35$ Over 10 Repetitions

	Mean	Standard deviation	Minimum	Maximum
HD vector	.0497	.0010	.0488	.0517
K-modes	.0654	.0026	.0639	.0703
AutoClass	.1438	.0161	.1304	.1737

classification accuracy. Obviously, this computational complexity is linear in n, the same as in the other two algorithms. It is noticeable that because our algorithm sequentially reduces the sample size n, this complexity is actually an upper bound, and the actual complexity will be steadily lowered in the process of clustering. In addition, the HD vector algorithm practically will take less run time because it does not have any convergence requirements.

To illustrate the foregoing discussions concerning computational complexity for the three algorithms, we report the run times (in seconds) of the data analyses given in Sections 4.2 and 4.3 (Tables 4 and 5). For the K-modes and HD vector algorithms, the actual run times of the full algorithm executions were recorded, whereas for the AutoClass, only the run time of one full iteration cycle was recorded. This is because the AutoClass algorithm involves human intervention in the course of execution; that is, the algorithm stops after a full execution cycle and requests that users respond as to whether to continue on another iteration cycle. With no way to precisely estimate the time length of the human response, we decide to report only the run time required before the algorithm stops the first time for the users' manual input.

In the comparison, we programmed the HD vector and K-modes algorithms in C++, the same programming language used by the AutoClass algorithm. We ran the three algorithms 10 times in the same PC with an Intel Pentium 4 processor with a 2.53-Hz CPU and 512 MB. The summary statistics of the run times are listed in Tables 4 and 5.

These results clearly indicate that the proposed HD vector algorithm was the fastest and the AutoClass algorithm was the slowest in both data analyses. Note that the actual run time of the AutoClass algorithm was in fact much longer in the data analyses, because more than one iteration cycle may be necessary. The superiority of the HD vector algorithm was very striking in the zoo data analysis, being on average four times faster than the K-modes algorithm and 78 times faster than the one-cycle AutoClass algorithm. But this advantage was not as dramatic in the soybean data analysis, being only 30% faster than the K-modes algorithm and two times faster than the AutoClass algorithm. This is reflected by the fact that the HD vector algorithm sequentially reduces the sample size and thus is significantly beneficial in clustering a large dataset over its two competitors.

Table 5. Summary of the Run Times (in seconds) for the Zoo Data Analysis With $n = 101$ and $p = 16$ Over 10 Repetitions

	Mean	Standard deviation	Minimum	Maximum
HD vector	.0022	.0001	.0020	.0023
K-modes	.0098	.0018	.0088	.0140
AutoClass	.1716	.0128	.1627	.2005

6. CONCLUDING REMARKS

The HD vector algorithm developed in this article is a distance function-based clustering method that uses a modified chi-squared statistic to determine underlying clusters. The proposed algorithm is particularly useful for analyzing relatively high-dimensional data of moderate size. We conducted some additional simulations, not reported here, to see how well the proposed algorithm performs for low-dimensional categorical data. We found that in most cases the proposed HD vector algorithm outperformed the AutoClass and the K-modes algorithms.

At the outset of clustering, this modified chi-squared test is used to verify whether there are any clustering patterns in the data. The proposed algorithm does not require any model assumptions for attributes or any expensive numerical optimization procedures. Because the algorithm extracts clusters sequentially (one cluster at each iteration), it does not need any convergence criterion. Moreover, because the algorithm is built on the basis of the contrast between the observed HD vector and the UHD vector, the modified chi-squared statistic depends on the distance function only in an indirect way. At the end, the method automatically determines the number of clusters, and the resulting partition of the data is unique regardless of the order of data input. Through simulated and real-world datasets, the HD vector algorithm appeared to be superior over the popular AutoClass and the K-modes algorithms.

Besides the AutoClass algorithm, there are other model-based clustering methods for estimating the number of clusters. For example, Banfield and Raftery (1993) and Fraley and Raftery (1998) proposed using the Bayes information criterion as the criterion for cluster selection. However, their methods are suitable only for data with continuous attributes. Most model-based algorithms are built on the framework of mixture models, for which the EM algorithm is often the method of choice for parameter estimation. Because the EM algorithm is known to be very slow, such model-based algorithms are not computationally efficient, especially for large datasets with high dimensions. Moreover, numerical optimization often encounters the problem of multiple local maxima, which requires a tremendous computational effort to find the global maximum. Thus the final clustering depends on the choice of initialization. In contrast, the proposed HD vector algorithm is computationally far less burdensome. Furthermore, most clustering algorithms require multiple scans of the dataset to achieve convergence, as pointed out by Bradley et al. (1998). By applying the scalable framework proposed by Bradley et al. (1998), our algorithm requires only one scan of the available dataset. To be more specific, we can fill the RAM buffer with data points. We apply the HD vector algorithm to the data points in the buffer and then detect significant cluster centers and extract the corresponding clusters (*compression set*). Data points that are not deleted remain in the RAM buffer (*retained set*), and the buffer is refilled to reach the full capacity. The process is terminated when all data points have been scanned and no more cluster centers are detected. This feature is very attractive, especially when multiple scans of extremely large datasets with high dimensions are prohibitively expensive.

The modified chi-squared statistic plays a central role in the proposed HD vector algorithm, which follows approximately

a chi-squared distribution. Obviously, this asymptotic distribution may not work well when the sample size is too small or when in the chi-squared statistic a certain frequency at an individual distance value is too low. However, if the sample size is small, then perhaps a direct analysis rather than using a sophisticated clustering algorithm can give satisfactory results without much effort. Moreover, the latter case, where one or more individual terms in the modified chi-squared statistic contain very few observations, would rarely occur. This is because Theorem 2 implies that the first several components of the HD vector algorithm with respect to the cluster center should contain some decent numbers of data points if clusters exist in the data.

Although the HD vector algorithm is designed to analyze data with categorical attributes, the basic idea is also applicable to cluster data with continuous attributes. A natural extension of the proposed algorithm is to handle data of mixed types, with some categorical attributes and some continuous attributes. Other important future work is to extend the proposed method by incorporating missing values on some attributes. As seen in the soybean disease data, we ignored those data points involving missing values in the analysis. Currently we are investigating how the imputation strategy may be incorporated with the chi-squared test based on the HD vectors.

APPENDIX: PROOFS

Proof of Theorem 1

Let $\mathbf{X}_{n\times p}$ be the categorical dataset. Then, for any position, $\mathbf{S} = (s_1, s_2, \ldots, s_p)^T$, we have

$$\begin{aligned} D(\mathbf{S}, \mathbf{X}) &= \sum_{i=1}^{n} d(\mathbf{S}, \mathbf{X}_i) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{p} d_j\bigl(s_j, X_i^{[j]}\bigr) \\ &= \sum_{j=1}^{p}\sum_{i=1}^{n} d_j\bigl(s_j, X_i^{[j]}\bigr). \end{aligned}$$

Observe that $\sum_{i=1}^{n} d_j(s_j, X_i^{[j]})$ is the number of mismatches between the position s_j and each value of $\mathbf{Y}_j = (X_1^{[j]}, \ldots, X_n^{[j]})^T$. It follows from the definition of the cluster center that $\sum_{i=1}^{n} d_j(s_j, X_i^{[j]}) \geq \sum_{i=1}^{n} d_j(c_j, X_i^{[j]})$, where c_j is the jth component of the center position $\mathbf{C} = (c_1, c_2, \ldots, c_p)^T$. Therefore, c_j must be the most frequent state of $\mathbf{Y}_j$, which is the one that minimizes $\sum_{i=1}^{n} d_j(s_j, X_i^{[j]})$.

Proof of Theorem 2

By Definition 2, a center $\mathbf{C}$ minimizes (3). Observe that

$$\sum_{i=1}^{n} d(\mathbf{C}, \mathbf{X}_i) = \sum_{q=0}^{p} qU_q(\mathbf{C})$$

where

$$U_q(\mathbf{C}) = \sum_{i=1}^{n} I[d(\mathbf{X}_i, \mathbf{C}) = q], \qquad q = 0, 1, \ldots, p.$$

Hence $\mathbf{C}$ also minimizes the expected distance, $\mu_d(\mathbf{S}) = n^{-1} \times \sum_{q=0}^{p} qU_q(\mathbf{S})$.

Proof of Theorem 3

Consider uniformly distributed data, $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)$. For an arbitrary position $\mathbf{S} \in \Omega_p$, the HD vector $\mathbf{U}(\mathbf{S})$ consists of $(p+1)$ components, with the qth component given by

$$U_q(\mathbf{S}) = \sum_{i=1}^{n} I[d(\mathbf{X}_i, \mathbf{S}) = q], \qquad q = 0, 1, \ldots, p.$$

Because all data points are independent and identically distributed with an equal probability of being any position on the space Ω_p,

$$E\{U_q(\mathbf{S})\} = \sum_{i=1}^{n} P[d(\mathbf{X}_i, \mathbf{S}) = q] = n\frac{U_q^*(\mathbf{S})}{M}, \qquad q = 0, 1, \ldots, p,$$

where $M = \prod_{j=1}^{p} m_j$ is the total number of possible outcomes and $U_q^*(\mathbf{S})$ is the number of possible outcomes that have an exact distance q to the reference position $\mathbf{S}$ given by

$$\begin{aligned} U_0^* &= 1, \\ U_1^* &= (m_1 - 1) + (m_2 - 1) + \cdots + (m_p - 1), \\ U_2^* &= (m_1 - 1)(m_2 - 1) \\ &\quad + (m_1 - 1)(m_3 - 1) + (m_2 - 1)(m_3 - 1) \\ &\quad + \cdots \\ &\quad + (m_1 - 1)(m_p - 1) + (m_2 - 1)(m_p - 1) + \cdots \\ &\quad + (m_{p-1} - 1)(m_p - 1) \\ &= \sum_{i<j}^{p} (m_i - 1)(m_j - 1), \\ &\vdots \\ U_p^* &= (m_1 - 1)(m_2 - 1)\cdots(m_p - 1). \end{aligned}$$

Obviously, $U_q^*, q = 0, 1, \ldots, p$, do not depend on $\mathbf{S}$.

Proof of Theorem 4

Without loss of generality, we assume that all n data points are distinct. It suffices to show that for a given distance value q, the sum of the frequencies $U_q(\mathbf{S})$ over all of the positions in the space Ω_p equals the constant $M\varepsilon_q$, where M is the total number of positions in Ω_p, that is,

$$\sum_{\mathbf{S}\in\Omega_p} U_q(\mathbf{S}) = M\varepsilon_q. \tag{9}$$

In fact, the identity relation (9) is valid simply because

$$\begin{aligned} \sum_{\mathbf{S}\in\Omega_p} U_q(\mathbf{S}) &= \sum_{i=1}^{n}\sum_{\mathbf{S}\in\Omega_p} I[d(\mathbf{X}_i, \mathbf{S}) = q] \\ &= nU_q^* \\ &= M\varepsilon_q, \end{aligned}$$

where U_q^* is as given in Theorem 3. This identity relation (9) implies that there exists at least one position, say $\mathbf{S}_q^*$, at which the frequency $U_q(\mathbf{S}_q^*)$ must be bigger than ε_q, unless all of the terms in the sum are identically equal to ε_q, which is the case of no clustering pattern.

Proof of Theorem 5

An application of the Lagrange multiplier method leads to the following objective function:

$$G(U_{r+1},\ldots,U_p,\lambda) = \sum_{j=0}^{r}\frac{(U_j(\mathbf{S})-\mathcal{E}_j)^2}{\mathcal{E}_j} + \sum_{j=r+1}^{p}\frac{(U_j(\mathbf{S})-\mathcal{E}_j)^2}{\mathcal{E}_j} + \lambda\left(n-\sum_{j=0}^{p}U_j(\mathbf{S})\right).$$

Taking the first-order derivative of function G with respect to $U_j(\mathbf{S}), j = r+1,\ldots,p$, we obtain

$$\frac{\partial G(U_{r+1},\ldots,U_p,\lambda)}{\partial U_j} = \frac{2(U_j(\mathbf{S})-\mathcal{E}_j)}{\mathcal{E}_j} - \lambda = 0, \qquad j = r+1,\ldots,p.$$

It immediately follows that

$$\lambda = \frac{2(\sum_{j=0}^{r}U_j(\mathbf{S}) - \sum_{j=0}^{r}\mathcal{E}_j)}{\sum_{j=r+1}^{p}\mathcal{E}_j}$$

and

$$U_j(\mathbf{S}) = \left(\frac{\lambda}{2}+1\right)\mathcal{E}_j, = \frac{(\sum_{h=0}^{r}\mathcal{E}_h - \sum_{h=0}^{r}U_h(\mathbf{S}))\mathcal{E}_j}{\sum_{h=r+1}^{p}\mathcal{E}_h}, \qquad j = r+1,\ldots,p.$$

Plugging the $U_j(\mathbf{S}), j = r+1,\ldots,p$, into the modified chi-squared statistic, we obtain

$$\chi^2_M(r;\mathbf{S}) = \sum_{j=0}^{r}\frac{(U_j(\mathbf{S})-\mathcal{E}_j)^2}{\mathcal{E}_j} + \sum_{j=r+1}^{p}\frac{\{(1+\lambda/2)\mathcal{E}_j-\mathcal{E}_j\}^2}{\mathcal{E}_j}$$

$$= \sum_{j=0}^{r}\frac{(U_j(\mathbf{S})-\mathcal{E}_j)^2}{\mathcal{E}_j} + \frac{(\sum_{j=0}^{r}\mathcal{E}_j - \sum_{j=0}^{r}U_j(\mathbf{S}))^2}{\sum_{j=r+1}^{p}\mathcal{E}_j}.$$

This completes the proof.

[Received January 2004. Revised February 2005.]

REFERENCES

Anderberg, M. R. (1973), *Cluster Analysis for Applications*, New York: Academic Press.

Banfield, J., and Raftery, A. (1993), "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics*, 49, 803–821.

Bradley, P., Fayyad, U., and Reina, C. (1998), "Scaling Clustering Algorithms to Large Databases," in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, New York, August 1998, CA: AAAI Press, pp. 9–15.

Cheeseman, P., and Stutz, J. (1995), "Bayesian Classification (AUTOCLASS): Theory and Results," in *Advances in Knowledge Discovery and Data Mining*, eds. U. Fayyad, G. Piatesky-Shapiro, P. Smyth, and R. Uthurusamy, Menlo Park, CA: AAAI Press, pp. 153–180.

Fraley, C., and Raftery, A. (1998), "How Many Clusters? Which Clustering Method? Answer via Model-Based Cluster Analysis," *The Computer Journal*, 41, 578–587.

Gabarro-Arpa, J., and Revilla, R. (2000), "Clustering of a Molecular Dynamics Trajectory With a Hamming Distance," *Computers & Chemistry*, 24, 693–698.

Gąsieniec, L., Jansson, J., and Lingas, A. (2004), "Approximation Algorithms for Hamming Clustering Problems," *Journal of Discrete Algorithms*, 2, 289–301.

Gordon, A. D. (1999), *Classification*, London: Chapman & Hall.

Huang, Z. X. (1997), "Extensions to the *K*-Means Algorithm for Clustering Large Data Sets With Categorical Values," *Data Mining and Knowledge Discovery*, 2, 283–304.

Kaufman, L., and Rousseeuw, P. J. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, New York: Wiley.

Laboulais, C., Ouali, M., Le Bret, M., and Gabarro-Arpa, J. (2002), "Hamming Distance Geometry of a Protein Conformational Space. Application to the Clustering of a 4 ns Molecular Dynamics Trajectory of the HIV-1 Intergrase Catalytic Core," *Proteins: Structure, Function and Genetics*, 47, 169–179.

MacQueen, J. B. (1967), "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of the 5th Symposium at Mathematical Statistics and Probability*, Berkeley, CA: University of California Press, pp. 281–297.

Macwilliams, F. J. (1978), *The Theory of Error-Correcting Codes*, Amsterdam: North-Holland.

Ralambondrainy, H. (1995), "A Conceptual Version of the *K*-Means Algorithm," *Pattern Recognition Letters*, 16, 1147–1157.