Towards Finding Optimal Partitions of Categorical Datasets

Keke Chen Ling Liu
College of Computing, Georgia Institute of Technology
{kekechen, lingliu}@cc.gatech.edu

Technical Report, October, 2003

Abstract

A considerable amount of work has been dedicated to clustering numerical data sets, but only a handful of categorical clustering algorithms are reported to date. Furthermore, almost none has addressed the following two important cluster validity problems: (1) Given a data set and a clustering algorithm that partitions the data set into k clusters, how can we determine the best k with respect to the given dataset? (2) Given a dataset and a set of clustering algorithms with a fixed k, how to determine which one will produce k clusters of the best quality? In this paper, we investigate the entropy and expected-entropy concepts for clustering categorical data, and propose a cluster validity method based on the characteristics of expected-entropy. In addition, we develop an agglomerative hierarchical algorithm (HierEntro) to incorporate the proposed cluster validity method into the clustering process. We report our initial experimental results showing the effectiveness of the proposed clustering validity method and the benefits of the HierEntro clustering algorithm.

1 Introduction

Data clustering is an important method in data analysis. A considerable amount of work has been done in clustering numerical datasets. Most traditional clustering techniques define distance functions, such as Euclidean distance, for any pair of items and then to group the items that are close into a cluster [14]. When a distance function is given, it is natural to introduce the density-based methods [7, 4] to clustering. Correspondingly, such distance functions and the density concept are also heavily used in cluster validity methods [13, 11].

Traditionally, clustering techniques are not directly applicable for categorical data. Preprocessing is commonly used to obtain the numeric features from categorical data for clustering. For example, in information retrieval, the vector model is applied where the frequency of occurrence of a word in document is used as a numerical feature for clustering. However, there are also many datasets containing categorical data, which could not be transformed to numerical features appropriately, where special categorical clustering is needed. Examples include partitioning market basket data to find localized association rules [1], DNA or protein sequence data in Bioinformatics, and alarm messages from intrusion detection systems. It is widely recognized that clustering based on the categorical features are useful in many application areas.

Categorical clustering has special characteristics compared to numerical clustering. First, since categories have no sequential meaning, the definition of distance between categorical records is not intuitive except Hamming distance, in which the distance is equal to the number of unmatched attributes between two records. Second, similarity based on bulk of records, e.g. "orderliness" of records, is more significant for clustering categorical data. For example, the "group" operation in relational database produces groups where the values of grouped attributes are identical in each group – thus they are totally "ordered", while categorical clustering allows to introduce impurity into each group, as long as each group still shows high "orderliness". Third, when non-distance similarity criterion is used for categorical data, the traditional cluster validity methods are not applicable anymore. We need appropriate clustering validity methods to determine the optimal number of clusters and to evaluate the quality of clusters as well.

Cluster validity methods [11] are typically used to evaluate the quality of the clusters produced by certain clustering algorithm. One main issue in cluster validity is to determine the optimal number of clusters that fits the data set best. In most numerical clustering algorithms, the structure of clusters can be evaluated by the geometry and density distribution of clusters. (A good example is to evaluate the clustering result of a 2D experimental data set by visualizing them to see if the clustering result matches the geometry and density distribution of points.) A good numerical clustering scheme gives satisfactory "compactness" within clusters and enough "dissimilarity" between clusters. However, since the distance functions are not suitable and unintuitive for categorical data, intuitive concepts like the geometry and density distribution are not appropriate in evaluating the quality of categorical clusters.

Although it is unnatural to define a distance function between categorical data or to use the statistical center of a group of categorical items, there are some algorithms, for example, K-Modes [12] algorithm and ROCK [10] algorithm, try

to fit the traditional clustering methods into categorical data. However, since the numerical similarity/distance function might not describe the categorical properties properly and intuitively, it leaves little confidence to the clustering result. Furthermore, almost none has addressed the following two important cluster validity problems: (1) Given a data set and a clustering algorithm that partitions the data set into k clusters, how can we determine the best k with respect to the given dataset? (2) Given a dataset and a set of clustering algorithms with a fixed k, how to determine which one will produce k clusters of the best quality?

With these problems in mind, in this paper we propose a novel categorical cluster validity method and a hierarchical algorithm (**HierEntro**) to incorporate this cluster validity method into the categorical clustering process. This development is based on a number of observations. First, we argue that in categorical datasets, the "orderliness (disorderliness)" of a set of records (or rows, items, instance vectors ...) can be captured by the concept of entropy. Second, we observe that there is a connection between clustering and entropy: clusters of similar points have lower entropy than those of dissimilar ones, which also implies that a group of points in different clusters should have larger entropy than those in one cluster. Thirdly, we argue that the expected entropy and its characteristics play critical roles in evaluating cluster quality and determining the optimal partitioning schemes. Concretely, an ideal partitioning leaves satisfactory orderliness within a cluster and merging any two clusters in the partitioning scheme should introduce significant disorderliness. The main idea of our cluster validity method is based on the special properties of increment of expected-entropy with the decreasing of number of clusters. It suggested us finding the optimal numbers of partitions by observing the "ideal validity graph". The **HierEntro** algorithm was then proposed to produce an ideal validity graph. The experiments show that HierEntro can produce high quality clusters and help to determine the optimal clustering schemes.

The rest of the paper is organized as follows. Section 2 briefly reviews the related work. Section 3 defines the entropy and expected-entropy concepts used in categorical clustering, explores some important properties, and proposes an incremental expected-entropy based validity method and algorithm HierEntro. Section 4 demonstrates that this approach is effective in clustering and validating the experimental datasets. Finally, we conclude our work in section 5.

2 Related Work

While many numerical clustering algorithms [13, 14] have been published, only a handful of categorical clustering algorithms appear in literature. The general statistical analysis of categorical data was introduced in [2]. Many ideas in categorical clustering were also derived from the similar concepts and algorithms in numerical clustering. For example, KModes algorithm [12] employs the similar ideas in KMeans [13] algorithm, where the distance function and the definition of mean are re-designed for categorical data. With proper selection of the initial modes, KModes algorithm can be quite effective and very fast due to the $O(N)$ complexity.

Whereas, ROCK [10] is an adaptation of agglomerative hierarchical clustering algorithm and graph-based algorithm [14], which heuristically optimizes a criterion function defined in terms of the number of "links" between tuples. Jaccard coefficient is used to define the similarity, which is then used to defined the links. This linkage-based approach is still based on the distance function, therefore, it is only applicable to the applications, where the distance function makes sense.

Gibson et al. introduce STIRR [9], an iterative algorithm based on non-linear dynamical systems. STIRR represents each attribute value as a weighted vertex in a graph. Starting with the initial conditions, the system is iterated until a "fixed point" is reached. when the fixed point is reached, the weights in one or more of the "basins" isolate two groups of attribute values on each attribute. Even though they proved this approach works for some experimental datasets with two partitions, the user may hesitate in using it due to the unintuitive working mechanism.

CACTUS [8] adopts the linkage idea from ROCK and names it "strong connection". However, the similarity is calculated by the "support". A cluster is defined as a region of attributes that are pair-wise strongly connected.Similarly, the concept of "support" or linkage is still indirect in defining the similarity of categorical data, and unnecessarily makes the clustering process complicated. As we suggested, similarity measures based on bulk of records, like entropy, are more suitable.

Cheng et al. [6] applied the entropy concept in numerical subspace clustering, and Coolcat [5] introduced the entropy concept into categorical clustering further. There is a connection between clustering and entropy: clusters of similar points have lower entropy than those of dissimilar ones, which also implies that points in different clusters should have larger entropy than those in one cluster. Coolcat is kind of similar to KModes. However, Coolcat assigns the item to a cluster that minimized the expected-entropy. Considering the cluster centers may shifting, a number of worst-fitted points will be re-clustered after a batch. Even though Coolcat approach introduces the entropy concept into its categorical clustering algorithm, Coolcat did not consider the problem of finding the optimal number of categorical clusters.

C. Aggarwal [1] demonstrated that localized associations are very meaningful to market basket analysis. To find the localized associations, they introduced a categorical clustering algorithm CLASD to partition the basket data. They defined a new similarity measure for a pair of transactions. CLASD is still a kind of traditional clustering algorithm – the special part is the definition of similarity function for categorical data.

Most of recent research in categorical clustering is focused on clustering algorithms. Surprisingly, there is no research concerning about the special validity methods for categorical datasets. We will introduce our validity methods for categorical datasets in the following sections.

3 Expected-Entropy Based Categorical Cluster Validity Method

In this section, we begin with the specific definition of entropy and expected-entropy that are used in our cluster validity method. The characteristics of expected-entropy are then explored to introduce the proposed cluster validity method. With the properties of expected-entropy, we show that the expected-entropy of the optimal partitioning monotonously increases when the number of clusters of the partitioning scheme decreases. "Validity graph" describes the incremental rate of expected-entropy between the neighboring optimal schemes. On an "ideal validity graph", we can easily find the candidates for the possible optimal number of clusters and "OpPlot" is used to locate them precisely. Finally, an approximately optimal algorithm in finding ideal validity graph is described and its implementation HierEntro is analyzed.

3.1 Definition of Entropy and Expected Entropy

In the following discussion, we define a dataset $\mathbb{S}$ as a table having d columns and N rows. Let A_j denote the j-th column. There are a limited number of distinct categorical values defined in $domain(A_j)$. A_j is conceptually different from $A_k (k \neq j)$. The i-th row can be represented as a *d*-dimensional vector$\langle a_{i1}, a_{i2}, \ldots, a_{id}\rangle$, where $a_{ij} \in domain(A_j), 1 \leqslant i \leqslant N$. Let $P(a_{i1}, a_{i2}, \ldots, a_{id})$ be the probability of the vector $\langle a_{i1}, a_{i2}, \ldots, a_{id}\rangle$ appearing in dataset $\mathbb{S}$. The classical definition of entropy can be described as:

$$E_0(\mathbb{S}) = -\sum_{a_{i1}\in A_1} \cdots \sum_{a_{id}\in A_d} P(a_{i1}, a_{i2}, \ldots, a_{id}) \log_2 P(a_{i1}, a_{i2}, \ldots, a_{id}) \tag{1}$$

where $P \log_2 P = 0$, if $P = 0$. Given a dataset $\mathbb{S}$, the probability of one record can be estimated by the frequency of its occurrence in the instance space. From equation 1, it is easy to inference that $E_0(\mathbb{S}) \leqslant 0$. Generally, if the records are in D categories, the entropy is in range of [0, $\log_2 D$]. The largest value is reached when the categories are uniformly distributed. Since we should consider the orderliness/disorderliness in each column rather than in record level when we perform categorical cluster analysis, let us narrow down the entropy definition onto the column level and average the column entropies as the final entropy for the dataset. As above discussion, the entropy of each column may falls onto different ranges [0, $\log_2 D_j$], where D_j is the number of elements in $domain(A_j)$. Since the columns having more categories should have higher probability in generating large entropy and thus may dominate the overall entropy by simply averaging them, we need to normalize the entropy of each column before averaging. We argue that using $1/\log_2 D_j$ as the normalization weight is appropriate. First of all, it normalizes each column-entropy to the range of [0, 1] eliminating the dominating factors. Secondly, this normalization slightly reduces the entropy contribution of the columns that have more categories, which makes the potentially more disordered ("noisy") columns not disturbing the underlying cluster structure.

$$E(\mathbb{S}) = \frac{1}{d}\sum_{j=1}^{d} E(A_j), \qquad E(A_j) = -\frac{1}{\log_2 D_j}\sum_{a_{ij}\in A_j} P(a_{ij}) \log_2 P(a_{ij}) \tag{2}$$

where $P(a_{ij})$ denotes the probability of a categorical value a_{ij} in the j-th column and can be estimated by counting the occurrence frequency of the categorical value a_{ij} in j-th column. When a value a_{ij} in $domain(A_j)$ does not appear in the dataset $\mathbb{S}$, $P(a_{ij})$ is defined as 0. Obviously, we have $0 \leqslant \mathbb{S} \leqslant 1$.

Evaluating the cluster quality of a clustering scheme should consider the intra-quality of all clusters produced by the scheme. Partitions in different size having the same entropy should not be regarded as equivalent in the level of disorderliness. Here, we derive another important concept, the *expected entropy* (**EE**), for assessing the intra-cluster quality of a bunch of categorical clusters. Suppose a dataset is partitioned to *k* clusters. Let C^k denote a partitioning scheme of k clusters, $\mathbb{C}_i$ be the cluster i, which has entropy of $E(\mathbb{C}_i)$. Since large partitions have more influence over the quality of scheme, we define the expected entropy of a clustering scheme as the entropy sum of all partitions weighted by the size of each partition.

$$EE(C^k) = \sum_{i=1}^{k} \frac{N_i}{N} \cdot E(\mathbb{C}_i) = \frac{1}{N}\sum_{i=1}^{k} N_i \cdot E(\mathbb{C}_i) \tag{3}$$

where N is the total number of instances in the datasets, and N_i is the number of instances in *i*-th partition. We name the variable part $N_i \cdot E(\mathbb{C}_i)$ as the "weighted entropy", which shall be used in the following lemma proof.

Expected entropy could then be used to assess the intra-cluster quality of a categorical clustering scheme as the compactness indices do to the numerical clustering. We have the relationship: *The smaller expected-entropy $\Longleftrightarrow$ the possibly smaller entropy in each cluster $\Longleftrightarrow$ the more items similar in each clusters $\Longleftrightarrow$ the more "compact" the cluster*.

Therefore, given a bunch of clustering schemes with the same number of clusters, the best scheme can be defined as the one having the lowest expected-entropy.

Expected-entropy gives a way to assess the quality of clustering schemes if the number of cluster k is given. When k is not determined, we need to find the optimal k (or possibly a number of good k's), which gives acceptable quality among all clustering schemes. Traditionally, statistical indices are applied to evaluate the schemes with different number of clusters. The peaks, valleys, or distinguished knots on the index curve are regarded the interesting points, the corresponding number of clusters to which is then recommended as the best number of clusters (the best k). Can the expected-entropy serve as such an index? In order to answer this question, we have to explore the characteristics of the expected-entropy.

3.2 Characteristics of Expected-entropy in Finding the Globally Optimal Categorical Clusters

Given the number of clusters k, there is at least one optimal clustering scheme having the lowest expected-entropy among all possible k-cluster schemes. We would like to check over the expected-entropy curve of these optimal schemes. To discover the shape of expected-entropy curve, we start with the initial scenario where each instance is a cluster and the expected-entropy is zero (Lemma 1). We then use Lemma 2 to prove that the expected-entropy of optimal scheme monotonously increases when the number of clusters decreases (Lemma 3). So expected-entropy is inadequate to find the optimal k. However, the increasing rate of expected-entropy between the neighboring optimal schemes may reveal some important properties behind the expected-entropy curve. We then define the "validity graph" and "ideal validity graph" for probing the possible optimal schemes. To begin with, we first introduce the lemmas that help us describe the expected-entropy curve.

Lemma 1. $(\textbf{Zero Start})$*The smallest expected-entropy is zero when each instance is considered as a cluster.*

Proof : Since the expected-entropy is non-negative by the definition, the minimum value, zero, is reached when each cluster has zero entropy. This happens when each cluster contains identical instances and the extreme situation is that each cluster contains only one instance.

Any *k*-cluster scheme can be formed by merging two clusters of some (*k+1*)-cluster scheme, therefore, we are interested in the property of this merging.

Lemma 2. $(\textbf{Incremental Merging})$ *Merging two clusters in any clustering scheme increases the expected-entropy or keeps the expected-entropy unchanged.*

Proof : Intuitively the merging should increase the global disorderliness. Formally, let $\mathbb{C}_1$ and $\mathbb{C}_2$ be the two clusters to be merged, with N_1 and N_2 records respectively in a k-cluster partitioning scheme. By the definition of expected-entropy (equation 3), we want to prove that the sum of the weighted entropy of $\mathbb{C}_1$ and $\mathbb{C}_2$ is less than the weighted entropy of $\mathbb{C}_1 \cup \mathbb{C}_2$, i.e.

$$N_1 \cdot E(\mathbb{C}_1) + N_2 \cdot E(\mathbb{C}_2) \leqslant (N_1 + N_2)E(\mathbb{C}_1 \cup \mathbb{C}_2), \qquad or$$

$$-\frac{1}{d}\sum_{j=1}^{d}\frac{1}{\log_2 D_j}\sum_{a_{ij}\in A_j} N_1 \cdot P_1(a_{ij})\log_2 P_1(a_{ij}) - \frac{1}{d}\sum_{j=1}^{d}\frac{1}{\log_2 D_j}\sum_{a_{ij}\in A_j} N_2 \cdot P_2(a_{ij})\log_2 P_2(a_{ij})$$

$$\leqslant -\frac{1}{d}\sum_{j=1}^{d}\frac{1}{\log_2 D_j}\sum_{a_{ij}\in A_j}(N_1+N_2)\cdot P_{1\cup 2}(a_{ij})\log_2 P_{1\cup 2}(a_{ij}) \tag{4}$$

Therefore, to prove (4), we can check if the following relation is satisfied:

$$N_1 \cdot P_1(a_{ij}) \cdot \log_2 P_1(a_{ij}) + N_2 \cdot P_2(a_{ij}) \cdot \log_2 P_1(a_{ij}) \geqslant (N_1 + N_2)P_{1\cup 2}(a_{ij})\log_2 P_{1\cup 2}(a_{ij}) \tag{5}$$

Without loss of generality, suppose cluster $\mathbb{C}_1$ has k items and cluster $\mathbb{C}_2$ has m items having value a_{ij} at j-th column. The formula 5 can be transformed to $k\log_2\frac{k}{N_1} + m\log_2\frac{m}{N_2} \geqslant (k+m)\log_2\frac{k+m}{N_1+N_2}$. Since k, m, N_1, N_2 are positive integers, let $k = u \cdot m$ and $N_2 = v \cdot N_1, (u, v > 0)$, and then we can eliminate $\log_2$ to get a simpler form: $\frac{v}{(1+v)^{u+1}} \leqslant \frac{u^u}{(1+u)^{u+1}}$. It is easy to prove that $\frac{u^u}{(1+u)^{u+1}}$ is the maximum value of the function $f(v) = \frac{v}{(1+v)^{u+1}}(u, v > 0)$. Therefore, formula (5) is true and thus (4) is true. Lemma 2 is proved. We then try to see what the relationship is between the neighboring optimal schemes having k and $k+1$ clusters respectively.

Definition 1. *A k-cluster clustering scheme is optimal, if the scheme gives the minimal expected-entropy among all of the k-cluster schemes. We use $OpEE(C^k)$ to denote the minimal expected-entropy of k-cluster schemes, e.g. $OpEE(C^k) = min\{EE(C_i^k)\}$, where C_i^k represents any of the possible n-cluster clustering schemes.*

Lemma 3. *$OpEE(C^n) \geqslant OpEE(C^m)$, when $n < m$*

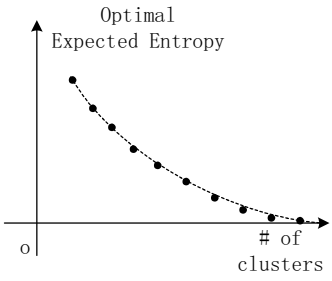

Figure 1: Sketch of expected-entropy graph.

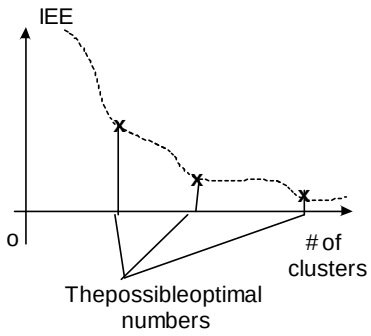

Figure 2: Sketch of ideal validity graph.

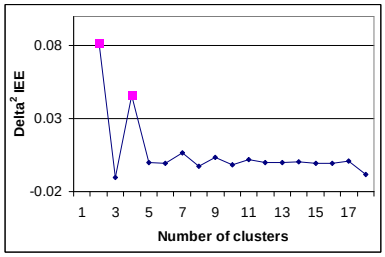

Figure 3: Finding optimal numbers (example of soybean-small data).

Proof : This lemma says that, when $n < m$, the optimal partitioning of n clusters has the same or larger expected-entropy than that of m clusters. It is easy to constructively prove it, when we have Lemma 2. Let us construct one arbitrary m-cluster scheme C_1^m by splitting the clusters in an optimal n-cluster scheme. With Lemma 2 and definition 1, we have

$$OpEE(C^n) \geqslant EE(C_1^m) \geqslant OpEE(C^m). \tag{6}$$

The first equation in above formula is satisfied when only merging of identical instance vectors happens from C_1^m to C^n. The second equation is satisfied when C_1^m is an optimal scheme of m clusters. Therefore, the optimal expected-entropy monotonically decreases with the increasing of the number of clusters. Our experimental results show that a reasonable categorical clustering algorithm produces expected-entropy curves similar to Figure 1. This curve implies that, 1) it is highly possible that the globally optimal clustering scheme is not unique in terms of expected-entropy and 2) expected-entropy is not a good index for finding the optimal k.

What can we explore more from the expected-entropy relationship? Let us look at the incremental rate of expected-entropy instead. Intuitively, small incremental rate between two neighboring schemes implies that they are similar in terms of the expected-entropy. If expected-entropy increases a lot when the number of clusters is reduced, this reduction of number of clusters introduces too much disorderliness into the partitions and probably should not be suggested. By looking at the incremental rate of expected-entropy, we can probably find a series of neighboring "stable" schemes, which have small increasing rate on expected-entropy, and we may also find the points where a series "stable" schemes become a "less stable" scheme – the incremental rate increases dramatically. All of these changes on the incremental rates have some specific meaning behind.

Hence, in order to explore the possible globally optimal schemes, we first formally define *the increasing rate of expected-entropy* **(IEE)** as the increment of expected-entropy from the $(k+1)$-cluster clustering scheme to k-cluster scheme: $IEE(k) = EE(C^k) - EE(C^{k+1})$, and the IEE plot as the "**Validity Graph**". We also define a validity graph satisfying $IEE(k) \geqslant IEE(k+1)$ for $k = 1..N-1$, is an "**Ideal Validity Graph (IVG)**". Since an ideal validity graph is usually a step-like curve (Figure 2), it is much easier to find the candidates of globally optimal schemes from it – the global optimal schemes may be at the points of the ideal validity graph where the IEE drops sharply and reaches a "platform" as the number of clusters increases. Figure 2 sketches an ideal validity graph and the possible globally optimal schemes. Generally, the decision rules can be described as follows: *1) schemes are at the platform area of IVG* $\Longleftrightarrow$ *schemes have similar quality* and *2) a dramatic dropping at IVG implies a candidate of optimal scheme*

In order to observe the possible optimal numbers clearly, we try to extract the key "*knots*" on the ideal validity graph automatically. As above description, the optimal points can be defined as follows. k is an optimal point at the ideal validity graph, if $IEE(k-1) - IEE(k)$ is much larger than $(IEE(k) - IEE(k+1))$ compared to the other points in a local area $[k-n, k+n]$, where n is a small integer. These "knots" can be precisely extracted from the 2nd order differential of IEE curve. We define $\Delta IEE(k) = IEE(k) - IEE(k+1)$ and $\Delta^2 IEE(k) = \Delta IEE(k-1)$ - $\Delta IEE(k)$. By looking at the $\Delta^2 IEE(k)$ plot, we can find the candidates of optimal numbers are among the top-k peaks (Figure 3) from left to right. We name the plot of $\Delta^2 IEE(k)$ as "Optimal Top-k Plot (**OpPlot**)", which is only used to extract the candidates of optimal numbers from ideal validity graph automatically.

3.3 an Algorithm for Generating Ideal Validity Graph

The precise way to find the validity graph is to calculate the optimal expected-entropy for each k, $k = 1..N$. However, this optimal method is highly complex in computational cost for even a small dataset. To simplify the computation, we propose a greedy algorithm, which produces near optimal expected-entropy for each k and an ideal validity graph for identifying the candidates of the globally optimal schemes in an acceptable cost.

This algorithm begins with the scenario, where each instance vector is a cluster and the expected entropy is 0 as Lemma 1 describes, and then iteratively find a pair of clusters to merge, the merging of which leaves the minimum

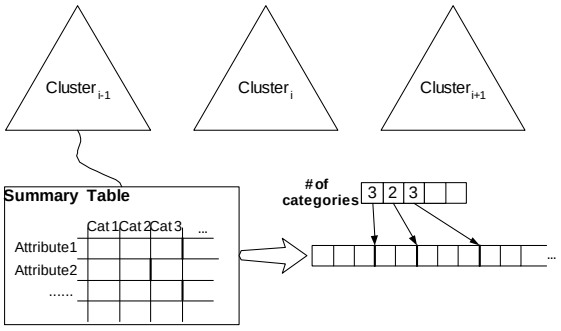

Figure 4: Summary table and physical structure

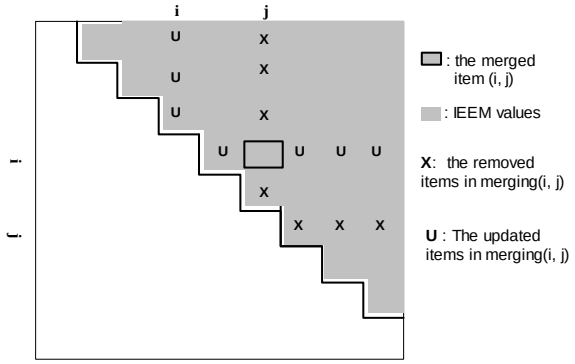

Figure 5: IEE table and the operation schedule following a merging operation

increment to the expected-entropy. The merging continues until the number of clusters becomes 1. The expected-entropy in each step is saved for generating validity graph and OpPlot. Meanwhile, the clusters and their sub-clusters form a cluster tree in the merging process and record the merging process precisely. Any of the candidate optimal schemes can be easily extracted from the tree and the hierarchical structure between the candidates also helps the user to understand the relationship between them. This merging does not ensure the optimality, however, the quality of the generated clustering schemes is proved very high in the experiments. More importantly, this algorithm could produce an ideal validity graph while the other algorithms do not. We shall prove it later.

We define *the incremental rate of expected-entropy in merging two clusters* (**IEEM**) as the dissimilarity measure for a pair of clusters. Concretely, if two clusters contain identical records, merging them will keep the expected-entropy unchanged (Lemma 2), thus $IEEM = 0$, and the dissimilarity is the lowest. If the two clusters are very different, the merging will introduce large disorderliness, i.e. $IEEM$ will be large and the dissimilarity between them is large.

Definition 2. *Let $\mathbb{C}_i$ and $\mathbb{C}_j$ denote the two merged clusters, having N_i and N_j records respectively. The incremental rate of expected-entropy in the merging: $IEEM(\mathbb{C}_i, \mathbb{C}_j) = \frac{1}{N}\{(N_i+N_j)\cdot E(\mathbb{C}_i \cup \mathbb{C}_j) - N_i \cdot E(\mathbb{C}_i) - N_j \cdot E(\mathbb{C}_j)\}$. Similarly, we define $IEEM(\mathbb{C}_i, \mathbb{C}_j, \mathbb{C}_k)$ as the incremental rate of expected-entropy in merging three clusters in the sequence of C_i and C_j and then C_k.*

With Lemma 2, we know $IEEM(\mathbb{C}_i, \mathbb{C}_j) \geqslant 0$. By the above definition we also have $IEEM(\mathbb{C}_i, \mathbb{C}_j, \mathbb{C}_k) = IEEM(\mathbb{C}_i, \mathbb{C}_j) + IEEM(\mathbb{C}_i \cup \mathbb{C}_j, \mathbb{C}_k) = \frac{1}{N}\{(N_i+N_j+N_k)E(\mathbb{C}_i \cup \mathbb{C}_j \cup \mathbb{C}_k) - N_i \cdot E(\mathbb{C}_i) - N_j \cdot E(\mathbb{C}_j) - N_k \cdot E(\mathbb{C}_k)\}$. Therefore, changing the sequence of merging three clusters will not change the increment of expected-entropy. Obviously, we also have $IEEM(\mathbb{C}_i, \mathbb{C}_j, \mathbb{C}_k) \geqslant IEEM(\mathbb{C}_x, \mathbb{C}_y)$, where (x, y) is any combination of two elements in (i, j, k).

Lemma 4. *The greedy algorithm generates an ideal validity graph.*

Proof: sorting any pair of clusters $(\mathbb{C}_i, \mathbb{C}_j)$ by $IEEM(\mathbb{C}_i, \mathbb{C}_j)$ value in ascending order, the greedy algorithm merges the pair having the minimum $IEEM(\mathbb{C}_i, \mathbb{C}_j)$ each time, and the global weighted expected-entropy is increased by the minimum $IEEM(\mathbb{C}_i, \mathbb{C}_j)$ at meanwhile. After merging, the $IEEM(\mathbb{C}_i, \mathbb{C}_k)$ and $IEEM(\mathbb{C}_j, \mathbb{C}_k)(k \neq i, j)$ should be updated to $IEEM(\mathbb{C}_i, \mathbb{C}_j, \mathbb{C}_k)$, since the cluster $\mathbb{C}_i$ and $\mathbb{C}_j$ have been replaced by the cluster $\mathbb{C}_i \cup \mathbb{C}_j$. As we have discussed, we have the updated value $IEEM(\mathbb{C}_i, \mathbb{C}_j, \mathbb{C}_k) \geqslant IEEM(\mathbb{C}_i, \mathbb{C}_j)$. Therefore, in next merging the greedy algorithm will pick a pair of clusters, which, in any case, has $IEEM$ value greater than $IEEM(i, j)$. Generated by this algorithm, the incremental rate of expected-entropy between two neighboring schemes $IEE(k)$ is equal to the $IEEM$ at step k. Therefore, the produced validity graph is an ideal validity graph.

At first look, it seems that finding the pair of clusters of minimum $IEEM$ is very costly. However, with the proper data structure, the entire algorithm can be implemented in the complexity of $O(N^2)$ $IEEM$ calculation, plus $O(N^2)$ heap operations (insertions and deletions).

3.4 The Hierarchical Expected-entropy Based Clustering/ Validating Algorithm (HierEntro)

We implement the algorithm proposed in the last section with an efficient hierarchical way (**HierEntro**). "Summary tables" and a priority queue built on heap are used to simplify the $IEEM$ calculations and the merging operations.

The expected-entropy computation is based on frequency counting of each categorical value in each attribute. To facilitate the frequency counting process, we build up a summary table for each cluster to keep track of the frequency of each category in each attribute. Such a summary table is also helpful for the merging operation. To merge two clusters, we just need to sum up the two summary tables as that of the new cluster.

A priority queue built on heap is used to choose the candidate pair clusters having minimum $IEEM$. An auxiliary 2D table ($IEEM$-table) (Figure 5) is used to save the IEEM value of merging each pair of clusters. Initially the table has less than half entries ($\frac{N(N-1)}{2}$) full and each entry is pushed into the priority queue.

The merging operation is not only merging two summary tables, but also updating the related $IEEM$ entries. Concretely, if C_i is the master cluster in merging (C_i, C_j), C_j should be merged to C_i and the C_j related $IEEM$ entries, i.e. the entries (k, j) and $(j, k)(k \neq j)$ should be removed from the priority query. And the entries (i, k) or $(k, i)(k \neq i)$ should be updated with the value $IEEM(C_i, C_j, C_k)$.

The algorithm **HierEntro** output a set of expected-entropies indexed by the number of clusters, and the hierarchical cluster label tree which encoding all candidate optimal schemes. Due to the space limitation, we give the concrete algorithm in the appendix.

3.5 Complexity Analysis for HierEntro Algorithm

The cost of HierEntro comes from several parts, including the initialization of summary tables, the initialization of $IEEM$-table, the merging operations, and the heap operations. Initializing summary table costs $O(N)$ I/O operation and Initializing $IEEM$-table costs $O(N^2)IEEM$ calculation. Each $IEEM$ calculation includes adding up the two summary tables and computing the weighted entropy with the merged summary tables. However, the calculation of weighted entropy can be simplified, since:

$$\begin{aligned} N_i \cdot E(C_i) &= -\frac{1}{d}\sum_{j=1}^{d}\frac{1}{\log_2 D_j}\sum_{\substack{a_{jk}\in A_j \\ c_{jk}=freq(a_{jk})}} N_i \cdot \frac{c_{jk}}{N_i}\log_2\frac{c_{jk}}{N_i} \\ &= -\frac{1}{d}\sum_{j=1}^{d}\frac{1}{\log_2 D_j}\sum_{\substack{a_{jk}\in A_j \\ c_{jk}=freq(a_{jk})}} c_{jk}(\log_2 c_{jk} - \log_2 N_i) \end{aligned}$$

Since the frequency c_{jk} and the number of instance N_i are non-negative integers, an auxiliary array can be used to buffer the $\log_2$ values. By using the log_2 value buffer, the complexity of calculating weighted entropy can be reduced to $O(dm)$, where d is the number of attributes and m is the average number of categories in an attribute. The total runs of merging operations cost $O(N^2)IEEM$ calculation and $O(N^2)$ heap operations (deletions and insertions, each in $O(\log N)$). Therefore, the total time complexity is $O(dmN^2 + N^2 \log N)$.

The summary tables require $O(dmN)$ space, the auxiliary array for log calculation costs $O(N)$ space and both of $IEEM$-table and the heap cost $O(N^2)$. So totally, the algorithm needs $O(N^2)$ space to process a categorical dataset in size of N rows by d columns.

4 Experimental Results

In this section, we first explore more characteristics of the proposed validity method with some small synthetic datasets for easy understanding. Then, we show the quality of HierEntro clustering results on two real datasets: archaeological and soybean-small data, compared to the other three categorical clustering algorithms: ROCK [10], KModes [12] and CoolCat [5], in terms of expected-entropy. Validity graphs on the above results show how the validity method with HierEntro works in revealing the possible optimal partitioning schemes of real categorical datasets. The experimental results conclude that HierEntro can not only gives clustering schemes in high quality, but also help to determine the possible globally optimal clustering schemes.

Another problem with the categorical clustering is how to understand the clustering result well. Interpreting the result with external labels is a good way helping to understand the clustering results. Previous research on categorical clustering falsely used the external labels to validate the quality of their algorithmic results. However, the cluster structure is determined only by the inherent properties of the dataset with the meaningful similarity measures, not by any external information. Therefore, we would like to name this validation process as "*validating the consistency between external labels and the defined cluster structure*", rather than "*validating the quality of clusters with external labels*". The proposed validity method, together with HierEntro, can help to check the consistency efficiently.

4.1 Experimental characteristics of the validity method

Let us construct a 2-column 10-record categorical dataset DS1: {(a 0), (a 1), (b 0), (b 1), (c 0), (c 1), (d 0), (d 1), (e 0), (e 1)}. The first column contains 5 categories (a, b, c, d, e) equally in size and the second column contains boolean values (0,1) equally in size. The ideal validity graph (Figure 6) suggests that the partitioning of 2 clusters, which equally partitions the dataset by the boolean column, is the best one. Roughly, the ideal validity graph shows the schemes from 2-cluster to 8-cluster are all very similar in terms of expected-entropy, so we can choose the 2-cluster scheme as the best. People may intuitively think that another scheme of 5 clusters, i.e. the one partitioned by the first column, is a suboptimal one, but the graph seems not suggest a 5-cluster scheme, but a 4-cluster scheme. What is the difference here? Let us look

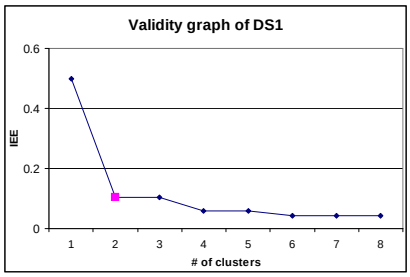

Figure 6: Ideal validity graph for DS1

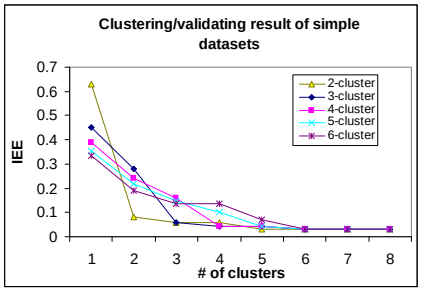

Figure 7: Validity graph of some simple datasets.

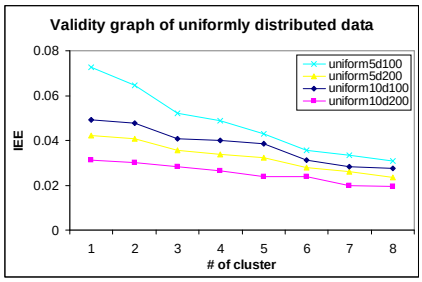

Figure 8: Validity graph for uniformly distributed data

at the 5-cluster schemes generated by HierEntro, {{(a 0), (b 0), (e 0)}, {(a 1), (e 1)}, {(b 1)}, {(c 0), (d 0)}, {(c 1), (d 1)}}. It seems that the partitioning is biased by the second column. This is true since the second column has categories and thus weighed more in expected-entropy. By doing so, HierEntro can correctly focus on the optimal schemes with less number of clusters.

Experiments on more datasets clearly show that HierEntro can capture the optimal numbers correctly. We design a set of 2-column 10-record datasets, where the first column contains a number of clusters, while the second column has 10 categories represented by 0..9, which do not suggest any cluster structure. For example, a 6-cluster dataset is (a 0), (a 1), (b 2), (b 3), (c 4), (c 5), (d 6), (d 7), (e 8), (f 9).

Datasets in uniform or normal distribution are tested to see if the validity results confirm the inherent structures. 4 different datasets for each distribution are used. Each column of the datasets contains 8∼12 categories. Uniform datasets have all categorical values in one column uniformly distributed. A normal-distribution dataset has one multi-dimensional center, and all categorical values are normally distributed around this center. so there is no non-trivial cluster structure(only one cluster) in the normal datasets either. The validity graphs (Figure 8 and 9) confirm that there is no obvious cluster structure in all of the datasets – no significant IEE dropping happens.

4.2 Finding the possible optimal partitions in real datasets

In this section, we want to experimentally prove that the HierEntro clustering results have very good quality in terms of expected entropy, even though it does not ensure the optimal expected-entropy. The validity graphs are also used to find the possible optimal partitions. Two datasets are used here.

- Archaeological data is used in Coolcat paper [5] and originally in [3]. This is a hypothetical collection of human tombs and artifacts from an archaeological site. It has 9 attributes and 20 instances. The first attribute indicates the sex (M for male, F for female) of the individuals buried. The other eight attributes are binary (1 present, 0 non-present), and represent artifacts types (e.g., ceramics, bracelets, arrow points) that were found (or not found) in the tomb. So basically it is a boolean dataset.

- Soybean-small dataset was used by K-Modes [12] and can be found in UCI machine learning database (www.ics.uci.edu). This is a small subset of the original larger data set(soybean-large). It has 35 attributes and 47 instances, containing 4 clusters as the document says. The 35 attributes contains the month, temperature, leaf shapes, seed conditions, and all other factors that can be significant in causing soybean diseases. There are 8 columns containing more than 2 categories, several columns having identical data (thus no use in clustering), and the others boolean data.

We choose to evaluate four algorithms: K-Modes, Coolcat, ROCK [10] and HierEntro. We will briefly introduce the compared three algorithms in the related work. Among the three, K-Modes and Coolcat are consistent with the expected-entropy oriented method, while ROCK is a distance/linkage based algorithm, which is not consistent with the expected-entropy criterion. The quality of different clustering schemes are evaluated by the expected-entropies. Since Coolcat is a randomized algorithm, we run Coolcat 20 times on each number of clusters and pick the best result as 'Coolcat-best' and the average of the 20 results as 'Coolcat-avg'.

The expected-entropy graph shows that the three algorithms: HierEntro, Coolcat and K-Modes produce results in similar quality, while ROCK presents not so good expected-entropies since it uses the linkage-based approach. Even though the results of 'Coolcat-best' are likely to reach the optimal expected-entropy, HierEntro almost always generates the schemes having the lowest expected-entropy among the three algorithms. In addition, only the validity graphs of HierEntro are ideal validity graphs, which can help us conveniently determine the optimal number of clusters. We can see that other algorithms give irregular zigzag lines indicating nothing significant for evaluation.

In the OpPlot of archaeological dataset, 2 is highly suggested as the optimal number of partitioning (Figure 14). Checking the produced labels for 2-cluster scheme, we find it is 100% consistent with the 'sex' column! This is interesting! For soybean dataset, 2-cluster or 4-cluster partitioning are recommended as the best ones (Figure 15). Checking the

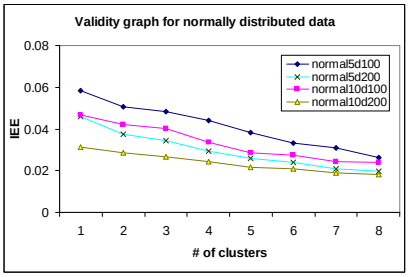

Figure 9: Validity graph for normally distributed data.

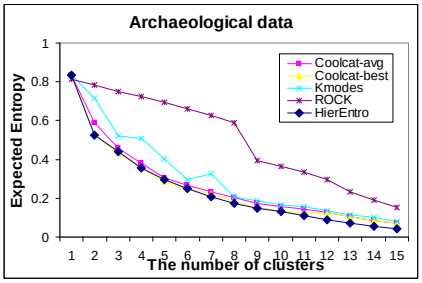

Figure 10: Expected-entropy of archaeological data.

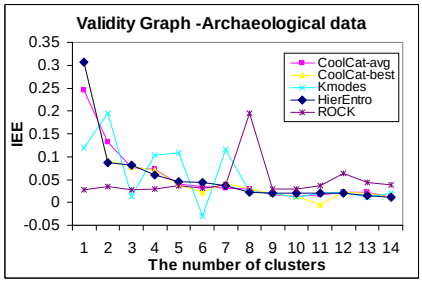

Figure 11: Validity graph of archaeological data.

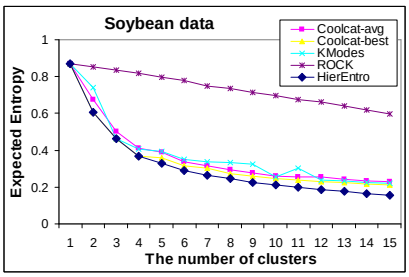

Figure 12: Expected-entropy of soybean data.

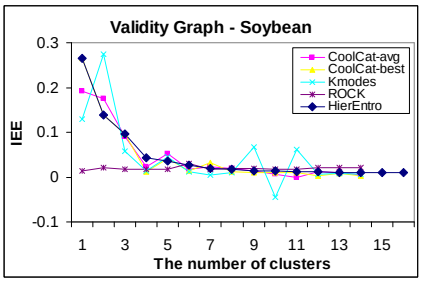

Figure 13: Validity graph of soybean data.

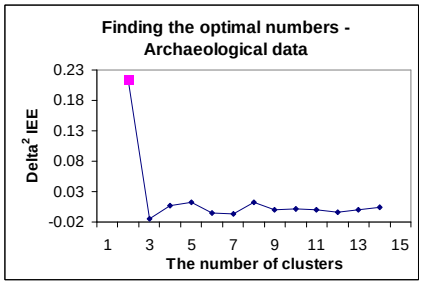

Figure 14: Finding the optimal numbers for archaeological data.

description given in the literature – there are 4 types of soybean diseases being tested, so we look at the 4-cluster clustering result. Comparing the external labels (labeling the datasets with the four diseases) to the HierEntro 4-cluster result, we surprisingly found that they are also 100% consistent! The experimental result on the real datasets proves that HierEntro and the validity method can be effective in identifying the interesting structure.

4.3 Validating the Consistency between External Labels and Entropy-based Clusters

Since clustering is an unsupervised grouping process, there is no explicit label provided for clustering. Any validity methods using external labels to validate the cluster quality are not feasible. However, external labels are often provided as the application specific knowledge by domain experts to explain or understand the clustering result. Re-categorizing the clustering result with the domain knowledge is also an efficient way to understand the inherent structure in the dataset. However, since there are many possible clustering schemes, how to choose the candidates for consistency checking is a important problem. The validity method plays important role in choosing the candidates, while the HierEntro result makes this process even easier – checking one candidate is enough to determine the consistency level.

We define the consistency between external labels and a clustering scheme as follows. Looking at one cluster, find the external label that labels the most cluster members and regard it as the major label. The other labels are regarded as minor labels for this cluster. The impurity of external labelling is evaluated by the percentage of the minor labels over all labels. Lower impurity means the corresponding external labels are more consistent with the categorical clustering result.

Since HierEntro merges a pair of cluster in each step, with the reduction of number of clusters, the impurity should increase or keep unchanged by the definition. Therefore, we do not need to check the consistency on each possible optimal number. Instead, we only check the impurity level around the largest possible optimal number of clusters to determine consistency level.

Let us look at the example of “mushroom” dataset. The external labels are given by the original dataset, indicating “poisonous” or “edible”. They were shown possibly consistent with the categorical clustering result by ROCK [10]. We checking the consistency one a 2000-record random sample. As the document of the dataset describes – there are around 20 kinds of mushrooms, we choose the largest optimal number of clusters (< 20) from the OpPlot, which is 16. The corresponding impurity plot shows that in partition number of 16 the unmatching rate is very low, only about 0.9%. Therefore, we confirm that the consistency level is high. Re-categorizing these 16 clusters to either “poisonous” or “edible” is meaningful. Further domain-related analysis can be focused on the clusters that are not purely labelled.

5 Conclusion and Future Work

Most of the recent research about categorical clustering is adapting the techniques in numerical clustering to the categorical counterpart. Since this revision, especially, the distance definition, is very unnatural for categorical data, the clustering

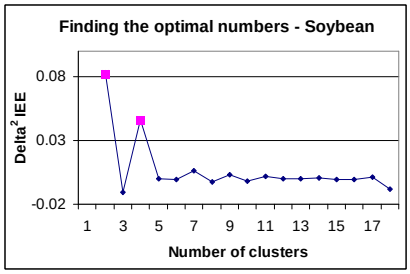

Figure 15: Finding the optimal numbers for soybean data.

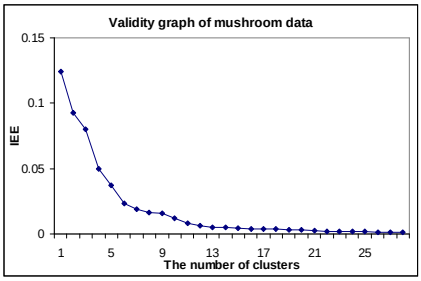

Figure 16: Validity graph of mushroom data.

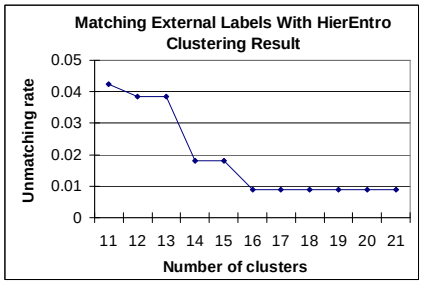

Figure 17: Unmatching rate of external labels.

results are not well-understandable, nor applicable in combining domain knowledge. In this paper, we proposed a cluster validity method based on the characteristics of entropy and expected-entropy. We developed an agglomerative hierarchical algorithm (HierEntro) to incorporate the proposed cluster validity method into the clustering process. The validity method suggests us to find optimal number of clusters by observing the ideal validity graph. The HierEntro algorithm then proposed to generate an ideal validity graph. Our proposed approach has three unique features: First, it produces high-quality clusters in terms of expected-entropy. Second, it recommends those globally optimal partitioning schemes to be the candidate choices to set the best value for the parameter k (the total number of clusters). These candidates are formed in a hierarchical way, making it easy to understand the correlation between the candidates. Thirdly, the HierEntro clustering algorithm helps to validate the consistency between the external labels and the inherent cluster structure efficiently.

From our initial experiments, we have shown the effectiveness of the proposed clustering validity method and the benefits of the HierEntro clustering algorithm. Although the current complexity $O(N^2 \log N + dmN^2)$ prevents HierEntro processing very large datasets efficiently, many techniques, such as sampling or summarization can be potentially combined to process very large datasets. Therefore, improving the scalability of HierEntro will be the next step towards finding optimal partitions of large categorical datasets.

References

[1] AGGARWAL, C. C., MAGDALENA, C., AND YU, P. S. Finding localized associations in market basket data. *IEEE Trans. Knowledge and Data Eng. 14*, 1 (2002), 51–62.

[2] AGRESTI, A. *Categorical Data Analysis*. Wiley-Interscience, 1990.

[3] ALDENDERFER, M., AND BLASHFIELD, R. *Cluster Analysis*. Sage Publications, 1984.

[4] ANKERST, M., BREUNIG, M. M., KRIEGEL, H.-P., AND SANDER, J. Optics: Ordering points to identify the clustering structure. *Proc. of ACM SIGMOD* (1999).

[5] BARBARA, D., LI, Y., AND COUTO, J. Coolcat: an entropy-based algorithm for categorical clustering. *Proc. of ACM Conf. on Information and Knowledge Mgt. (CIKM)* (2002).

[6] CHENG, C. H., FU, A. W.-C., AND ZHANG, Y. Entropy-based subspace clustering for mining numerical data. *Proc. of ACM SIGKDD* (1999).

[7] ESTER, M., KRIEGEL, H.-P., SANDER, J., AND XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Second International Conference on Knowledge Discovery and Data Mining* (1996).

[8] GANTI, V., GEHRKE, J., AND RAMAKRISHNAN, R. Cactus-clustering categorical data using summaries. *Proc. of ACM SIGKDD* (1999).

[9] GIBSON, D., KLEINBERG, J., AND RAGHAVAN, P. Clustering categorical data: An approach based on dynamical systems. *Proc. of VLDB 8*, 3–4 (2000), 222–236.

[10] GUHA, S., RASTOGI, R., AND SHIM, K. Rock: A robust clustering algorithm for categorical attributes. *Proc. of IEEE Intl. Conf. on Data Eng. (ICDE)* (1999).

[11] HALKIDI, M., BATISTAKIS, Y., AND VAZIRGIANNIS, M. Cluster validity methods: Part i and ii. *SIGMOD Record 31*, 2 (2002).

[12] HUANG, Z. A fast clustering algorithm to cluster very large categorical data sets in data mining. *Workshop on Research Issues on Data Mining and Knowledge Discovery* (1997).

[13] JAIN, A. K., AND DUBES, R. C. *Algorithms for Clustering Data*. Prentice hall, 1988.

[14] JAIN, A. K., AND DUBES, R. C. Data clustering: A review. *ACM Computing Surveys* (1999).