The "Best K" for Entropy-based Categorical Data Clustering

Keke Chen Ling Liu

Georgia Institute of Technology
{kekechen, lingliu}@cc.gatech.edu

Abstract

With the growing demand on cluster analysis for categorical data, a handful of categorical clustering algorithms have been developed. Surprisingly, to our knowledge, none has satisfactorily addressed the important problem for categorical clustering – how can we determine the best K number of clusters for a categorical dataset? Since the categorical data does not have the inherent distance function as the similarity measure, the traditional cluster validation techniques based on the geometry shape and density distribution cannot be applied to answer this question. In this paper, we investigate the entropy property of the categorical data and propose a *BkPlot* method for determining a set of candidate "best Ks". This method is implemented with a hierarchical clustering algorithm *HierEntro*. The experimental result shows that our approach can effectively identify the significant clustering structures.

keywords Categorical Data Clustering, Entropy, Cluster Validation

1 Introduction

Data clustering is an important method in data analysis. Clustering algorithms use the similarity measure to group the most similar items into clusters [23]. Clustering techniques for categorical data are very different from those for numerical data in terms of the definition of similarity measure. Most numerical clustering techniques use distance functions, for example, Euclidean distance, to define the similarity measure, while there is no inherent distance meaning between categorical values.

Traditionally, categorical data clustering is merged to numerical clustering through the data preprocessing stage [23], where numerical features are extracted/constructed from the categorical data, or the conceptual similarity between data records is defined, based on the domain knowledge. However, meaningful numerical features or conceptual similarity are usually difficult to extract at the early stage of data analysis because we have little knowledge about the data. It has been widely recognized that clustering directly on the raw categorical data is important for many applications. Examples include environmental data analysis [29], market basket data analysis [1], DNA or protein sequence analysis [8], and network intrusion analysis [5]. Therefore, there are increasing interests in clustering categorical data recently [21, 19, 17, 18, 6, 15, 3, 25].

Cluster Validation Different clustering algorithms hardly generate the same clustering result for one dataset, and we need the cluster validation methods to evaluate the quality of the clustering results [27, 22, 20]. Formally, there are two main issues in cluster validation: 1) how to evaluate the quality of different partition schemes generated by different clustering algorithms for certain dataset, given the fixed K number of clusters; 2) how to determine the numbers of clusters (the "best K"), which indicates the inherent significant clustering structures of the dataset.

For numerical data, the clustering structure is usually validated by the geometry and density distribution of the clusters. When a distance function is given for the numerical data, it is also natural to introduce the density-based methods [16, 4] into clustering. As a result, the distance functions and density concepts play the unique roles in validating the numerical clustering result. Various statistical cluster validation methods and visualization-based validation methods have been proposed for numerical data [22, 20, 12] and all are based on the geometry and density property. The intuition behind the geometry and density distribution justifies the effectiveness of these cluster validation methods. A good example commonly seen in clustering literature is evaluating the clustering result of 2D experimental datasets by visualizing it – the clustering result is validated by checking how well the clustering result matches the geometry and density distribution of points through the cluster visualization.

While lack of the distance meaning for the categorical data, the techniques used in cluster validation for numerical data are not applicable for categorical data. Without reasonable numerical feature extraction/construction for a given categorical dataset, the general distance functions are usually inapplicable and unintuitive. As a result, no geometry/density-based validation method is appropriate in validating the clustering result for categorical data.

Entropy Based Similarity Instead of using distance function to measure the similarity between any pair of data records, similarity measures based on the "purity" of a set of records seem more intuitive for categorical data. Entropy [14] is such a formal definition for measuring the purity of partition. Originally from information theory, Entropy has been applied in both pattern discovery [10] and numerical clustering [13]. Due to the lack of intuitive distance definition for categorical values, recently, there have been efforts in applying the entropy criterion in clustering categorical data [6, 25]. The initial results show that entropy criterion can be very effective in clustering categorical data. Li et al [25] also proved that the entropy criterion can be formally derived in the framework of probabilistic clustering models, which further supports that the entropy criterion is a

meaningful and reliable similarity measure for categorical data.

In entropy-based categorical clustering, the quality of clustering result is naturally evaluated by the entropy criterion [6, 25], namely, the *expected entropy* for a partition. However, the other cluster validation problem – determining the "best K", has not been sufficiently addressed yet. In this paper, we present a novel method based on entropy to address this problem.

Our Approach We first develop an agglomerative hierarchical clustering algorithm "*HierEntro*". The algorithm works in a bottom-up manner. Beginning with each individual record as a cluster, it merges the most similar pair of clusters in each step, where the similarity is evaluated with the *incremental entropy*. An agglomerative hierarchical clustering algorithm typically generates a clustering tree that contains the different clustering structures that have different K. We use these clustering structures to analyze the best K problem.

Based on the intuition behind the merging operation in HierEntro algorithm, we investigate the relation between the pairs of neighboring partition schemes (having K clusters and $K+1$ clusters, respectively). We use "*Entropy Characteristic Graph* (ECG) " to sketch the entropy property of the clustering structures, and use "*Best-K Plot* (BkPlot)", which is built on ECG, to identify the candidates of the best K. The initial experimental result shows that the proposed validation method, concretely, using the BkPlots generated by HierEntro to identify the best Ks, works effectively in finding the significant Ks for categorical data clustering.

The rest of the paper is organized as follows. Section 2 sets down the notations and gives the definition of the traditional entropy-based clustering criterion. Section 3 presents the agglomerative hierarchical clustering algorithm, HierEntro. Section 4 investigates the relation between the neighboring partitioning schemes with the entropy criterion, and proposes the validation method for indicating the best Ks. We present the experimental result in section 5 and review the related categorical clustering work in section 6. Finally, we conclude the paper in section 7.

2 Notations and Definitions

We give the notations used in this paper and then introduce the traditional entropy-based clustering criterion. Several basic properties about the entropy criterion will be presented later.

Consider that a dataset $\mathbb{S}$ with N records and d columns, is a sample set of the discrete random vector $X = (x_1, x_2, \ldots, x_d)$. For each component x_j, $1 \leqslant j \leqslant d$, x_j takes a value from the domain A_j. A_j is conceptually different from $A_k (k \neq j)$. There are a finite number of distinct categorical values in $domain(A_j)$ and we denote the number of distinct values as $|A_j|$. Let $p(x_j = v)$, $v \in A_j$, represent the probability of $x_j = v$, we have the classical entropy definition [14] as follows.

$$\begin{aligned} H(X) &= \sum_{j=1}^{d} H(x_j) \\ &= -\sum_{j=1}^{d} \sum_{v \in A_j} p(x_j = v) \log_2 p(x_j = v) \end{aligned}$$

Since $H(X)$ is estimated with the sample set $\mathbb{S}$, we define the estimated entropy as $\hat{H}(X) = H(X|\mathbb{S})$, i.e.

$$\begin{aligned} \hat{H}(X) &= H(X|\mathbb{S}) \\ &= -\sum_{j=1}^{d} \sum_{v \in A_j} p(x_j = v|\mathbb{S}) \log_2 p(x_j = v|\mathbb{S}) \end{aligned}$$

Suppose the dataset $\mathbb{S}$ is partitioned into K clusters. Let $C^K = \{C_1, \ldots, C_K\}$ represent a partition, where C_k is a cluster and n_k represent the number of records in C_k.

The classical entropy-based clustering criterion tries to find the optimal partition, C^K, which maximizes the following entropy criterion [9, 11, 25].

$$O(C^K) = \frac{1}{d}\left(\hat{H}(X) - \frac{1}{n}\sum_{k=1}^{K} n_k \hat{H}(C_k)\right)$$

Since $\hat{H}(X)$ is fixed for a given dataset $\mathbb{S}$, maximizing $O(C^K)$ is equivalent to minimize the item $\frac{1}{n}\sum_{k=1}^{K} n_k \hat{H}(C_k)$, which is named as the "expected entropy" of partition C^K. Let us notate it as $\bar{H}(C^K)$. For convenience, we also name $n_k \hat{H}(C_k)$ as the "weighted entropy" of cluster C_k.

Li et al [25] showed that the minimization of expected-entropy is equivalent to many important concepts in information theory, clustering and classification, such as Kullback-Leibler Measure, Maximum Likelihood [24], Minimum Description Length [26], and dissimilarity coefficients [7]. Entropy criterion is especially good for categorical clustering due to the lack of intuitive definition of distance for categorical values. While entropy criterion can also be applied to numerical data [13], it is not the best choice since it cannot describe the cluster shapes and other numerical clustering features of the dataset.

3 HierEntro Categorical Clustering Algorithm

In this section, we define the proposed similarity measure, *incremental entropy*, for two clusters. With incremental entropy, we design the algorithm HierEntro. HierEntro and its working mechanism is the tool used to explore the significant clustering structures in the next section.

3.1 Incremental Entropy

We investigate the mergence of two clusters to explore the similarity between any two clusters. Intuitively, merging the two clusters that are similar in the inherent structure will not increase the disorderliness (expected-entropy) of the partition, while merging dissimilar ones will inevitably bring larger disorderliness. Therefore, this increase of expected entropy has some correlation with the similarity between clusters. It is necessary to formally explore the property of merging clusters. Let $C_p \cup C_q$ represent the mergence of two clusters C_p and C_q, and C_p and C_q have n_p and n_q members, respectively. By the definition of expected entropy, the difference between $\hat{H}(K)$ $\hat{H}(K+1)$ is only the difference between the weighted entropies, $(n_p + n_q)\hat{H}(C_p \cup C_q)$ and $n_p \hat{H}(C_p) + n_q \hat{H}(C_q)$. We have the following relation for the weighted entropies.

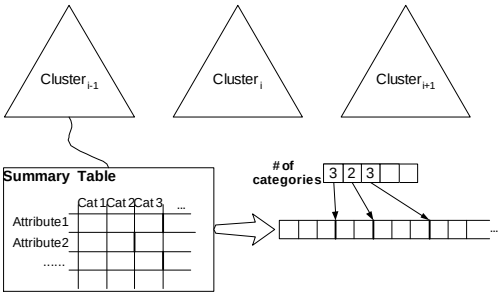

Figure 1: Summary table and physical structure

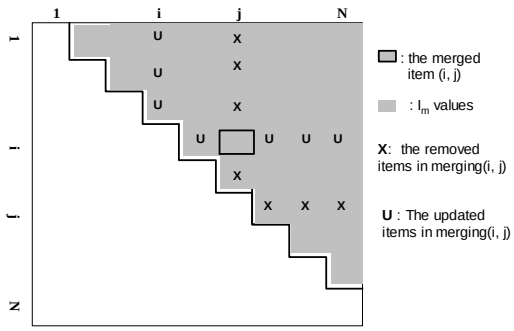

Figure 2: Illustration of the operation schedule after a merging operation

ds1	ds2
1 1 0 1	1 1 0 1
1 1 0 1	0 0 1 1
0 0 1 1	
0 0 1 1	

Table 1: Identical structure

Proposition 1. $(n_p + n_q)\hat{H}(C_p \cup C_q) \geqslant n_p\hat{H}(C_p) + n_q\hat{H}(C_q)$

PROOF. The result is quite intuitive and proving it is not hard (Appendix).

Let $I_m(C_p, C_q) = (n_p+n_q)\hat{H}(C_p \cup C_q) - (n_p\hat{H}(C_p) + n_q\hat{H}(C_q))$ be the "*incremental entropy*" by merging the clusters C_p and C_q. Note that $I_m(C_p, C_q) = 0$ most likely suggests that the two clusters have the identical structure – for every categorical value v_i in every attribute x_j, $1 \leqslant i \leqslant |A_j|$, $1 \leqslant j \leqslant d$, we have $p(x_j = v_i|C_p) = p(x_j = v_i|C_q)$. Identical structure implies $\hat{H}(C_p \cup C_q) = \hat{H}(C_p) = \hat{H}(C_q)$ regardless of the size of the clusters. A simple example in table 1 demonstrates the identical structure.

Incremental entropy brings the heuristic about the dissimilarity between any two clusters, i.e., when the two clusters are similar in structure, merging them will not bring large disorderliness into the partition, thus, $I_m(C_p, C_q)$ will be small; when the two clusters are very different, merging them will bring great disorderliness, thus, $I_m(C_p, C_q)$ will be large. Therefore, incremental entropy intuitively serves as the similarity measure between any two clusters.

3.2 HierEntro Algorithm

While the traditional hierarchical algorithms for numerical clustering needs to explicitly define the inter-cluster similarity with "single-link", "multi-link" or "complete-link" methods [22]. Incremental entropy is a natural cluster-based similarity measure, ready for constructing a hierarchical clustering algorithm. Having incremental entropy as the measure of inter-cluster similarity, we developed the following entropy-based agglomerative hierarchical clustering algorithm – (HierEntro).

HierEntro algorithm is a bottom-up process to construct a clustering tree. It begins with the scenario where each record is a cluster. Then, an iterative process is followed – in each step, the algorithm finds a pair of clusters C_p and C_q that are the most similar, i.e. $I_m(C_p, C_q)$ is minimum among all possible pair of clusters. We use $I_m^{(K)}$ to denot the I_m valude in forming K-cluster partition from K+1-cluster partition.

Maintaining the minimum incremental entropy for each step is the most costly part. In order to efficiently implement the HierEntro algorithm, we maintain three main data structures: *summary table* for convenient counting of occurrences of values, I_m*-table* for bookkeeping $I_m(C_p, C_q)$ of any pair of clusters C_p and C_q, and a I_m *heap* for maintaining the minimum I_m value in each step.

Summary table is used to maintain the fast calculation of cluster entropy $\hat{H}(C_k)$ and each cluster has one summary table (Figure 1). Since computing cluster entropy is based on counting the occurrences of categorical values in each column, a summary table keeps the counters for the cluster. Apparently, if the average column cardinality is m, a summary table keeps dm counters. Such a summary table enables fast merging operation – when merging two clusters, the two summary tables are added up to form the summary table for the new cluster.

We use I_m-table to keep track of the incremental entropy between any pair of clusters, which is then used to maintain the minimum-I_m for each round of mergence. The I_m-table is a symmetric table (thus, only a half of entries are used in practice), where the cell (i, j) keeps the value of $I_m(C_i, C_j)$ Figure 2.

I_m heap is used to keep track of the globally minimum incremental entropy. We define the most similar cluster of cluster u as

$$u.similar = \arg\min_v \{I_m(u, v), v \neq u\}$$

and let $u.I_m$ represent the corresponding incremental entropy of merging u and $u.similar$. We use $< u, u.I_m, u.similar >$ as the *feature vector* of cluster u. The feature vectors are inserted into the heap, sorted by $u.I_m$.

Algorithm 1 shows the sketch of the main procedure. When merging u and $u.similar$ happens, their summary tables are added up to form the new summary table. Consider u as the main cluster, i.e., $u.similar$ is merged to cluster u, we need to find the new $u.similar$ and insert the new feature vector $< u, u.I_m, u.similar >$ to the heap.

There is an important procedure for updating the bookkeeping information after merging operation. Let v denote the old $u.similar$. The bookkeeping information for v is released and any entries in I_m-table related to u or v should be updated. For any cluster c, if the $c.similar$ is changed due to the update of I_m-table, its location at the heap needs to be updated too. The detailed update algorithm is described in Algorithm 2 and demonstrated by Figure 2.

Algorithm 1 HierEntro.main()

$T_s[] \leftarrow$ initialize summary tables
$T_{I_m} \leftarrow$ initialize I_m table
$h \leftarrow$ heap
for Each record u **do**
  h.push $(< u, u.I_m, u.similar >)$
end for
while not empty(h) **do**
  $< u, u.I_m, u.similar > \leftarrow h.top()$
  $T_s[u] \leftarrow T_s[u] + T_s[u.similar]$
  update $< u, u.I_m, u.similar >$
  h.push $(< u, u.I_m, u.similar >)$
  updating_after_merging() //Algorithm 2
end while

Algorithm 2 HierEntro.updating_after_merging()

$C_i \leftarrow$ master cluster, $C_j \leftarrow$ merged cluster
release $T_s[C_j]$
invalidate I_m table entries $(C_j, *)$
update I_m table entries $(*, C_i)$ and $(*, C_j)$
for Each valid cluster u, if $u.similar == C_i$ or C_j **do**
  update $< u, u.I_m, u.similar >$;
  relocate $< u, u.I_m, u.similar >$ in h
end for

3.3 Complexity of HierEntro

Updating the I_m-table is the most costly part, consisting several incremental-entropy calculations. Each incremental-entropy calculation involves summation of the two summary tables and computing the weighted entropy with the merged summary tables. The cost of computing weighted entropy is O(dm), when an auxiliary array in length of N is used to buffer the $\log_2$ values as the following equation shows.

$$\begin{aligned} & n_p \hat{H}(C_p) \\ = & -\sum_{j=1}^{d} \sum_{\substack{v_{jk} \in A_j \\ c_{jk} = freq(v_{jk}) | C_p}} n_p \frac{c_{jk}}{n_p} \log_2 \frac{c_{jk}}{n_p} \\ = & -\sum_{j=1}^{d} \sum_{\substack{v_{jk} \in A_j \\ c_{jk} = freq(v_{jk}) | C_p}} c_{jk} (\log_2 c_{jk} - \log_2 n_p) \end{aligned}$$

The merge operations totally cost $O(N)$ incremental-entropy calculations but the total cost is dominated by updating I_m-table after each merging operation which will need $O(N^2)$ incremental-entropy calculations in total in the worst case. Therefore, the overall time complexity is $O(dmN^2)$. The summary tables require $O(dmN)$ space, both the log_2 buffer and the heap costs $O(N)$ space, and I_m-table costs $O(N^2)$ space.

We use the HierEntro algorithm as the tool to help understanding the property of significant clustering structures in categorical data. Having the expected entropy as the criterion of evaluating clustering quality for a fixed K, we will focus on the other important validation problem: what is the best Ks for a particular categorical dataset?

4 Exploring the Significant Clustering Structures

Traditionally, statistical validity indices based on geometry and density distribution are applied in clustering numerical data [20]. The statistical index values according to different K make an index curve. The Ks at the peaks, valleys, or distinguished "knees" on the index curve, are regarded as the candidates of the optimal number of clusters (the best Ks). Are there index curves indicating the significant clustering structures for categorical data too? Let $\bar{H}_{opt}(C^K)$ denote the expected entropy of the optimal partition of K clusters. The first thought might be investigating the curve of $\bar{H}_{opt}(C^K)$.

Our result shows that the curve of optimal expected-entropies is usually a smoothly decreasing curve without any distinguished peaks, valley, or knees (Figure 3). However, we find some special meaning behind the neighboring partition schemes (with K and $K+1$ clusters respectively). The differential of expected-entropy curve, which we name as "Entropy Characteristic Graph (ECG)" (Figure 4), has substantial meaning indicating the significant clustering structures. An ECG shows that the similar partition schemes with different K are at the same "plateau". From plateau to plateau there are the critical points implying the significant change of clustering structure, which can be the candidates for the best Ks. These critical points can be highlighted in the second-order differential of ECG, named "Best-K Plot (BkPlot)".

4.1 Property of Optimal Partition Schemes

In this section, we first give the Proposition 3 describing the relation between the optimal expected-entropies with varying K, which is then used to introduce the "Entropy Characteristic Graph" and "BkPlot".

Since the significant clustering structures are the globally optimal selections, we begin with the investigation of optimal partitions with varying K. Given the number K of clusters, there is at least one optimal partition minimizing the expected entropy $\bar{H}(C^K)$ – we name it as $\bar{H}_{opt}(C^K)$. There are several properties about $\bar{H}_{opt}(C^K)$.

First of all, $\bar{H}_{opt}(C^K)$ is bounded. It was proved in [25] that $\bar{H}(C^K)$ is bounded by $\hat{H}(X)$, i.e.

Proposition 2. $\hat{H}(X) \geqslant \bar{H}(C^K)$

$\bar{H}(C^K)$ is maximized when $K = 1$ – all data points are in the same cluster. We also have $\bar{H}(C^K) \geqslant 0$ as the entropy definition implies. The zero entropy $\bar{H}(C^k)$ is reached at $k = N$, when each vector is a cluster. Therefore, $\bar{H}_{opt}(C^K)$ is bounded by $[0, \hat{H}(X)]$.

Then, for any different number of clusters, K and L, $K < L$, we have also have the following property.

Proposition 3. $\bar{H}_{opt}(C^K) \geqslant \bar{H}_{opt}(C^L)$, *when* $K < L$

PROOF. Let some L-cluster partition C_0^L be formed by splitting the clusters in the optimal K-cluster partition. With Proposition 1, we have

$$\bar{H}_{opt}(C^K) \geqslant \bar{H}(C_0^L) \geqslant \bar{H}_{opt}(C^L)$$

□

Proposition 3 shows that the optimal expected-entropy decreases with the increasing of K, which meets the intuition well. It is hard to describe the curve with a closed form function. However, as our experimental result shows, it is often a negative logarithm-like curve (Figure 3). This curve implies that, 1) it is highly possible that the best K is not unique in terms of entropy criterion, and 2) expected-entropy curve could not help us to clearly identify the significant clustering structures.

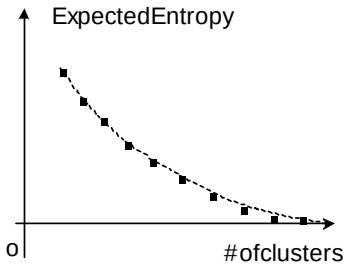

Figure 3: Sketch of expected entropy curve.

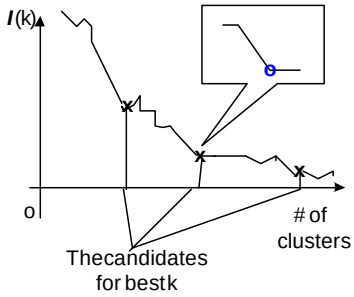

Figure 4: Sketch of ECG graph.

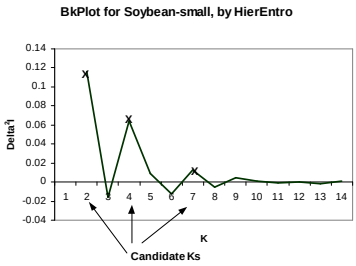

Figure 5: Finding the best k with BkPlot (example of soybean-small data).

4.2 Understanding the Similarity of Neighboring Partition Schemes

There is some important implication behind the expected-entropy curve when we consider the *similarity between the neighboring partitions* on the curve, where the neighboring partitions refer to the K-cluster partition and $K+1$-cluster partition. There are two aspects to capture this similarity. One aspect is the increasing rate of entropy, defined as $I(K) = \bar{H}_{opt}(C^{K+1}) - \bar{H}_{opt}(C^K)$, which indicates how much the clustering structure is changed. The other aspect is the difference between $I(K)$ and $I(K+1)$, which indicates whether the consecutive changes to the clustering structure are similar. Since it is hard to describe the relation between the optimal partitions, we use the cluster mergence described in HierEntro algorithm to intuitively illustrate the two aspects of similarity. In the consecutive partition schemes generated by HierEntro, the increasing rate is equivalent to incremental entropy: $I(K) = \frac{1}{Nd} I_m^{(K)}$.

First, we consider the meaning of small increasing rate of entropy. As we discussed, merging identical clusters introduces zero increasing rate, which implies that the merging does not introduce any impurity to the clusters and the clustering structure is not changed. Similarly, small increasing rate between two neighboring schemes implies that the reduction of number of clusters does not introduce large impurity to the partition and we consider the clustering structure is not significantly changed.

We can interpret the case of large increasing rate too. If the expected-entropy increases a lot from K+1 to K, this reduction of number of clusters should introduce considerable impurity into the partitions and thus the clustering structure can be changed significantly. In such cases, we need to investigate the relative changes in clustering structure of the neighboring schemes as follows.

Consider $I(K)$ as the amount of impurity introduced from K+1-cluster scheme to K-cluster scheme. If $I(K) \approx I(K+1)$, i.e. K-cluster scheme introduces similar amount of impurity as K+1-cluster scheme does, we regard that the clustering structure is not *relatively* changed from K+1-cluster scheme to K-cluster scheme. An example of "similar mergence" in Figure 6 can well demonstrate the similarity of clustering structure at $I(K) \approx I(K+1)$. We use icons to conceptually represent categorical clusters. The shape and the size of an icon represent the structure and size of the cluster, respectively. The four clusters ($C_1 \sim C_4$) in Figure 6 are very similar. They are selected in two consecutive merging operations. Thus, the changes to the resulting clustering structures are similar and not quite distinguishable from each other.

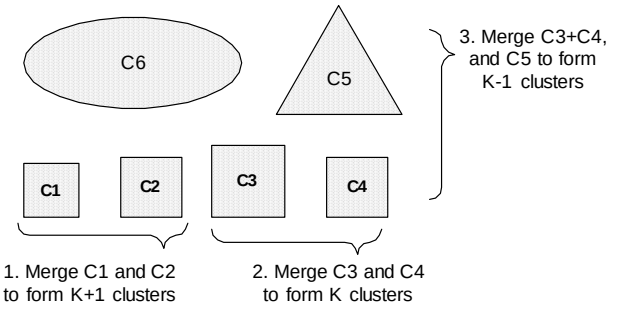

Figure 6: $I(K) \approx I(K+1)$, but $I(K-1) > I(K)$ significantly

However, the third merging operation, which merges $C_3 \cup C_4$ and C_5, might change the clustering structure greatly, and thus $I(K-1)$ can increase dramatically. This indicates that the second merge operation results in a representative clustering structure for cluster analysis.

In practice, if a dataset has significant clustering structure, we can find a series of neighboring "stable" schemes, which have similar increasing rate of entropy, and we may also find the *critical points* where a series of "stable" schemes become "less stable" – the increasing rate changes dramatically (Figure 4). Each of such critical points indicates some significant change in clustering structure and distinguishes a set of "stable" schemes from another set. All of the critical points should be the candidates for the best Ks and could be interesting to cluster analysis.

We name the $I(K)$ plot as *Entropy Characteristic Graph (ECG)*. If a dataset has significant clustering structures, its ECG should be a curve with some distinguished "knees". An ECG curve showing no distinguished knees implies that the clustering structure is smoothly changed when K changes from N to 1, and thus clustering structures at all Ks have the same importance – in other words, there is no significant clustering structure.

The common way to mathematically identify such critical knees on a curve is to find the peaks/valleys at the second-order differential of the curve. Since an ECG consists of a set of discrete points, we define the second-order differential of ECG as $\delta^2 I(K) - \delta I(K) = I(K) - I(K+1)$ and $\delta^2 I(K) = \delta I(K-1) - \delta I(K)$ to make K aligned with the critical points. We can clearly identify the best Ks at the $\delta^2 I(K)$ plot, and thus name it as the "Best-k Plot (BkPlot)" (Figure 5).

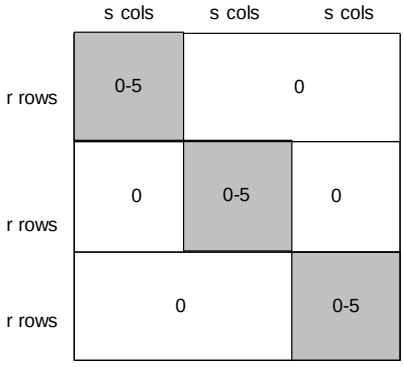

Figure 7: Synthetic Data DS1

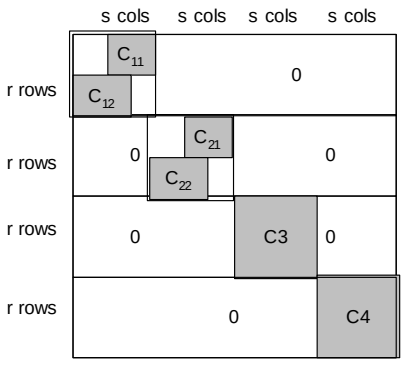

Figure 8: Synthetic Data DS2

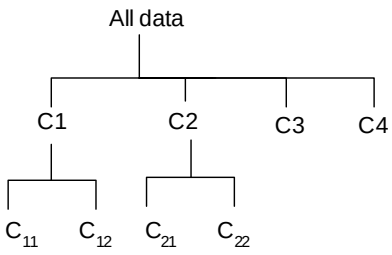

Figure 9: The significant clustering structures in DS2

4.3 Entropy Characteristic Graph Generated by HierEntro

ECGs generated by HierEntro have a special property. We use $I_m^{(K)}$ to denote the I_m value in forming K-cluster partition from $K+1$-cluster partition. Since $I(K) = \frac{1}{Nd} I_m^{(K)}$, it is equivalent to investigate the property of $I_m^{(K)}$. We will prove that $I_m^{(K)} \geqslant I_m^{(K+1)}$, so that the critical points always happen at the peaks of BkPlot.

Proposition 4. $I_m^{(K)} \geqslant I_m^{(K+1)}$

PROOF. Let $I_m(C_o, C_p, C_q)$ denote the incremental entropy in merging any three clusters. It is trivial to prove that the sequence of the three clusters does not matter in calculating the I_m and

$$I_m(C_o, C_p, C_q) \geqslant I_m(C_{(1)}, C_{(2)}) \tag{1}$$

where $C_{(1)}$ and $C_{(2)}$ are any two of the three clusters.

We maintain the ascending list of I_m for each merge operation in HierEntro algorithm. Suppose that the two clusters C_p and C_q are selected to merge to form the $K+1$-cluster scheme. We have $I_m^{(K+1)} = I_m(C_p, C_q)$. After the merge operation, the incremental entropy between the pairs of any cluster C_o, $o \neq p, q$, and the new cluster $C_p \cup C_q$, should be updated to $I_m(C_o, C_p, C_q)$. Since $I_m(C_p, C_q)$ is the minimum value at the stage $K+1$ and relation (1) shows the updates to I_m table only increase the values, the selected I_m value for stage K will definitely be greater or equal to that of stage $K+1$, i.e. $I_m^{(K)} \geqslant I_m^{(K+1)}$. □

The BkPlots of such ECGs ($I(K) \geqslant I(K+1)$) always exhibit the critical Ks at peaks. This could reduce the number of noisy Ks. We will demonstrate that the BkPlots generated by HierEntro are the most robust and efficient ones, compare to those generated by other algorithms.

5 Experimental Results

The goal of the experiments is twofold. 1)We want to show that BkPlot can be used to find the critical Ks. In order to precisely evaluate the effectiveness of the method, we design a set of datasets that have well-defined clustering structure. With these datasets, we can precisely compare the discovered clustering structures and the inherent clustering structures. 2) We want to show that the BkPlots generated by HierEntro are the most robust and efficient, compared to those by another two popular entropy-based clustering algorithms, Monte-Carlo method (MC) [25] and Coolcat [6] – all of the three algorithms try to minimize the expected-entropy defined in section 2.

5.1 Datasets

We construct two types of synthetic datasets with the following way, so that the clustering structure can be intuitively identified and manually labeled before running the experiments. The first type of datasets has a one-layered clustering structure (Figure 7) with 30 attributes and 1000 rows. It has three clusters with the same size. Each cluster has random categorical values selected from {'0','1','2','3','4', '5'} in a distinct set of attributes, while the rest attributes are set to '0'. The second one has a two-layered clustering structure also with 30 attributes and 1000 rows. The top layer has four clusters, two of which have sub-clusters as Figure 8 shows. Both types have clearly defined clustering structure, and each record in a generated dataset distinctly belongs to one cluster. We generate ten datasets for each type of structure, named DS1-i and DS2-i, $1 \leqslant i \leqslant 10$, respectively.

We also use three "real" datasets, "Soybean-small", "Congressional votes" and "Zoo" in the experiments. All of the three are from UCI KDD Archive [1].

- *Soybean-small data* is a dataset used to classify the soybean diseases. The dataset has 47 records and each record has 35 attributes describing the features of the plant. There are four classes in the dataset.
- *Congressional votes* is also a Boolean dataset containing US Congressional Voting Records for the year 1984. The dataset has 435 records. Each record has a Congressman's votes on 16 issues (i.e. 16 attributes). We use the 16 attributes to classify the Congressman to "Democrat" or "Republican".
- *Zoo data* contains the feature description of the animals in a zoo. There are 101 animal instances, classified to 7 categories. Each record has 17 attributes describing different features of animal, such as hair and the number of legs, most of which are boolean.

5.2 Compared Algorithms

Literally, any categorical clustering algorithm that employs the same entropy minimization criterion can generate a valid BkPlot. However, the quality of the BkPlots can be easily influenced by the underline algorithms. We briefly introduce another two algorithms, Monte-Carlo algorithm and Coolcat algorithm in this section. Both use expected entropy to evaluate the quality of partition and try to minimize the expected entropy in order to achieve a suboptimal

[1] http://www.ics.uci.edu/∼mlearn/MLRepository.html

partition. We compare the quality of BkPlots generated by the two algorithms to that by HierEntro.

Monte-Carlo Method [25] is a top-down partitioning algorithm. With a fixed K, it begins with all records in one cluster and follows an iterative process. In each step, the algorithm randomly picks one record from one of the K clusters and puts it into another randomly selected cluster. If the change of assignment does not reduce the expected entropy, the record is put back to the original cluster. The algorithm can be summarized as Algorithm 3.

Algorithm 3 Monte-Carlo Clustering

```
Input:( data records: X, # of clusters: K, # of unchanged steps: s)
Output: cluster assignment

Put all records into one cluster;
Calculate the initial expected entropy H_0;
Set the counter of unchanged steps, c ← 0;
while c < s do
  Randomly pick a point x from a cluster A;
  Randomly pick another cluster B;
  Put x into B, and calculate the new expected entropy H;
  if H > H_0 then
    Put x back to A, c ← c + 1;
  else
    H_0 ← H, c ← 0;
  end if
end while
```

Theoretically, given a sufficiently large s, the algorithm will eventually terminate at a near optimal solution. We set $s = 5000$ for running MC on the synthetic datasets.

To improve the efficiency, we also combine MC algorithm with the Coolcat algorithm, in practice. Instead of beginning with all records in one cluster, we use Coolcat algorithm to generate the initial partition, and then use MC algorithm to polish the partition, further reducing the expected entropy.

Coolcat [6] algorithm begins with selecting K records, which maximize the K-record entropy, from a sample of the dataset as the initial K clusters. It sequentially processes the rest records and assigns each to one of the K cluster. In each step, the algorithm finds the best fitted one of the K clusters for the new record – adding the new record to the cluster will result in minimum increase of expected entropy. The data records are processed in batches. Because the order of processing points has a significant impact on the quality of final clusters, there is a "re-clustering" procedure at the end of each batch. This procedure picks m percentage of the worst fitted records in the batch and re-assigns them to the K clusters in order to reduce the expected entropy further.

Algorithm 4 Coolcat Clustering

```
Input:( data records: X, # of clusters: K, re-clustering percent: m )
Output: cluster assignment

Find the K records as the initial clusters from the sample set, which
maximized the entropy of the K records;
for each batch do
  for each record u in the batch do
    Find the cluster C_i, putting u in which can result in the minimum
    incremental entropy;
    Place u in C_i;
  end for
  Find the worst fitted m percent of records in the batch;
  Re-clustering the worst fitted records;
end for
```

We run the algorithm on each dataset with a large sample size (50% of the datasets) and $m = 20\%$, which is sufficient for improvement through re-clustering [6]. In order to reduce the effect of ordering, we run Coolcat 20 times for each datasets and each run processes the data in a randomly generated sequence. Finally, we select the result having the lowest expected entropy among the 20 results.

5.3 Performance Measures

We use four measures to evaluate the quality BkPlots generated by different algorithms.

- *Coverage Rate.* We evaluate the robustness of BkPlot with "Coverage Rate (CR)" – how many significant inherent clustering structures are indicated by the BkPlot. There could be more than one significant clustering structures for a particular dataset. For example, four-cluster and six-cluster structures can be all significant for DS2. An robust BkPlot should always include all of the significant Ks.

- *False Discovery Rate.* There could be some Ks, which are actually not critical but suggested by some BkPlots. In order to efficiently find the most significant ones, we prefer a BkPlot to have less false indicators as possible. We use "*False Discovery Rate*(FDR)" to represent the percentage of the noisy indicators in the BkPlot.

- *Expected Entropy.* Since the BkPlot is indirectly related to expected entropy through ECG, it is also reasonable to check the quality of expected entropy for the partitions generated by different algorithms at the particular Ks. The most reliable BkPlot should be based on the expected entropy of optimal partitions for varying K. Because finding the optimal partitions is a NP-hard problem, we do approximation in all of the three algorithms. For a set of datasets in the same clustering structure, like DS1-i, $1 \leqslant i \leqslant 10$, we have almost same optimal expected entropy for different datasets at a fixed K. Using the mean-square-error (MSE) criterion [24] to evaluate the quality of the approximation result, we can decompose the errors to two parts: the deviation to the optimal expected entropy, and the variance of the estimated expected entropy. Let $\hat{h}$ be the estimated expected entropy and h be the optimal one. Let $E[\hat{h} - h]$ be the expected bias and $var(\hat{h})$ is the variance of $\hat{h}$.

  $$MSE = E^2[\hat{h} - h] + var(\hat{h})$$

  Without calculating the optimal expected entropy h, if an algorithm generates BkPlots with the lowest expected entropy and minimum variance among the three algorithms, we can also conclude that this algorithm is the best one of the three.

- *Purity.* For the real datasets, there is no documented clustering structure, but the class definition, which describes the domain knowledge, is given. We use $purity$ [30] to evaluate the consistency between the clustering result and the class definition. The purity of a cluster, $P(C_k)$, measures the extent to which the cluster contains data points primarily from a single

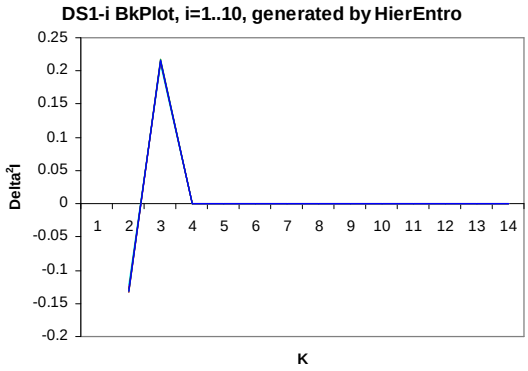

Figure 10: BkPlots of DS1 by HierEntro

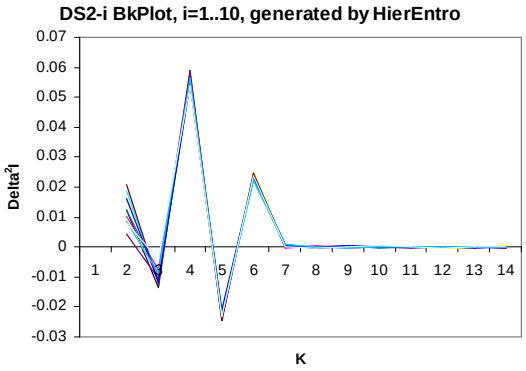

Figure 11: BkPlots of DS2 by HierEntro

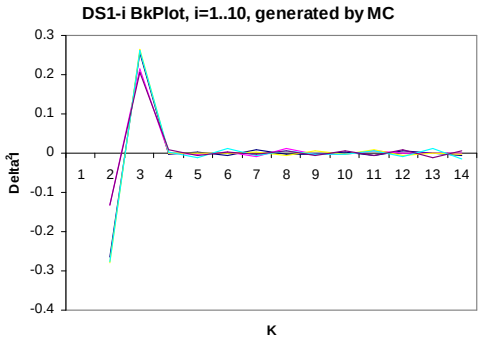

Figure 12: BkPlots of DS1 by MC

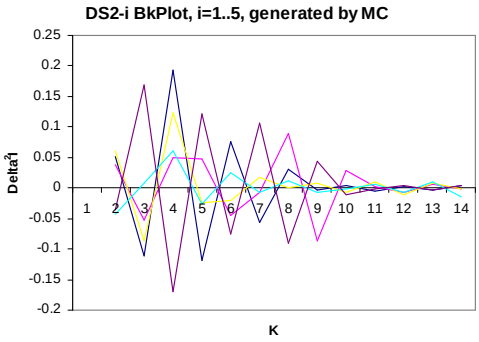

Figure 13: BkPlots of DS2 by MC

class. The purity of a clustering result is the weighted sum of the purity of individual cluster, given by

$$Purity = \sum_{k=1}^{K} \frac{n_k}{n} P(C_k)$$

5.4 Discussion

The BkPlots generated by HierEntro algorithm for DS1 (Figure 10 clearly indicate '3' is the only significant K. The datasets having the same clustering structure should have almost the identical BkPlots. The identical BkPlots on ten different DS1-i, $0 \leqslant i \leqslant 10$, shows that HierEntro is a robust algorithm for generating BkPlot.

The peaks of BkPlots for DS2-i (Figure 11) include the two inherent significant Ks – '4' and '6', but '2' is also given as the third significant K. However, we notice that the peak values at 'K=4' or 'K=6' for different DS2 datasets are almost same, while those at 'K=2' have more variation. This solicits us to consider a more reliable method to estimate the most significant K for a considerably large dataset. We can generate a bunch of sample sets, which have the identical clustering structure with the original dataset. The most stable peaks in the BkPlots of the sample sets correspond to the most significant Ks.

The BkPlots generated by Monte-Carlo algorithm for DS1 (Figure 12) also clearly identify that '3' is the best K with very small variation. However, the BkPlots for DS2 show large variation on Ks. In order to clearly observe the difference, we only show five BkPlots for DS2-i, $1 \leqslant i \leqslant 5$, respectively. Overall, the Ks distribute from '2' to '10' for different DS2-i. Some BkPlots include the significant Ks - '4' and '6', while others miss one or both, which implies that MC algorithm might not be robust enough for datasets having complicated clustering structure. The reason is MC algorithm becomes more likely to trap in local minima with the increasing complexity of clustering structure and increasing number of clusters.

Coolcat algorithm is the least robust one for generating BkPlots. It brings large variation for both datasets (Figure 14 and 15). Coolcat algorithm is originally designed for fast processing of categorical data while the quality of result is not well guaranteed. Therefore, it is not suitable for generating robust BkPlots.

We summarize the result with the discussed measures, Coverage Rate (CR), False Discovery Rate (FDR), and expected entropy (EE) in Table 2 and 3. The higher the coverage rate, the more robust the BkPlot is. The lower the false discovery rate the more efficient the BkPlot is. The numbers are the average over the 10 datasets. For both types of dataset, HierEntro shows the minimum expected entropy and minimum standard deviation, as well as the highest CR and lowest FDR. Therefore, the BkPlots generated by HierEntro are the most robust and efficient ones.

	CR	FDR	EE
HierEntro	100%	0%	0.732 ± 0.001
MC	100%	0%	0.733 ± 0.001
Coolcat	60%	85%	1.101 ± 0.026

Table 2: Summary for DS1-i

	CR	FDR	EE $K = 4$	EE $K = 6$
HierEntro	100%	33%	0.562 ± 0.002	0.501 ± 0.001
MC	80%	53%	0.565 ± 0.009	0.521 ± 0.008
Coolcat	60%	70%	0.852 ± 0.023	0.761 ± 0.021

Table 3: Summary for DS2-i

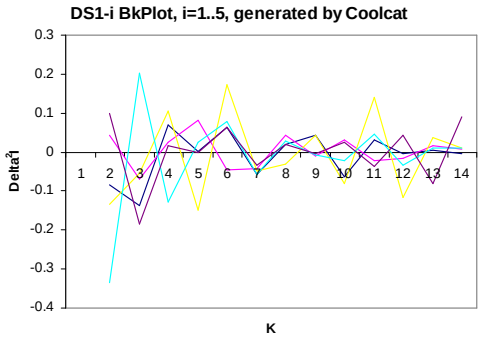

Figure 14: BkPlots of DS1 by Coolcat

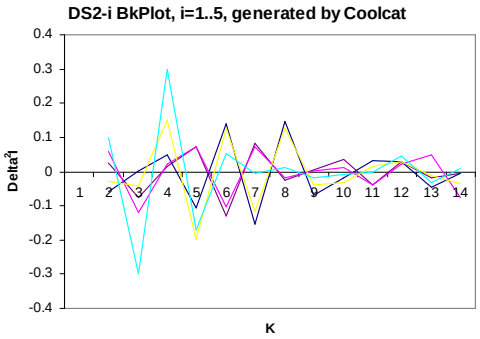

Figure 15: BkPlots of DS2 by Coolcat

dataset	N	d	# class	Best Ks	Purity
soybean-small	47	35	4	{2,4,7}	100%
votes	435	16	2	{2}	83%
zoo	101	17	7	{2,4,7}	93.1%

Table 4: HierEntro result for real datasets

We run experiments on real datasets with HierEntro only and the results match the domain knowledge very well. We are not clear about the best K for the inherent clustering structure, but we can use the documented number of classes as the reference number. Interestingly, the BkPlots of HierEntro shows that these numbers are all included in the best Ks, which implies that the inherent structure is consistent with the domain knowledge. In fact, the additional best Ks can be investigated further to explore more hidden knowledge. For example, 'K=2' and 'K=4' for zoo dataset might be other meaningful categorizations for the animals. The high purity also shows that the entropy-based categorical clustering can generate results highly consistent with the domain knowledge, which have been supported by other literatures [6, 25]. The result encourages us to believe that BkPlots with HierEntro can also work effectively for the real datasets.

6 Related Work

While many numerical clustering algorithms [22, 23] have been published, only a handful of categorical clustering algorithms appear in literature. The general statistical analysis of categorical data was introduced in [2]. Although it is unnatural to define a distance function between categorical data or to use the statistical center (the mean) of a group of categorical items, there are some algorithms, for example, K-Modes [21] algorithm and ROCK [19] algorithm, trying to fit the traditional clustering methods into categorical data. However, since the numerical similarity/distance function may not describe the categorical properties properly and intuitively, it leaves little confidence to the clustering result.

Gibson et al. introduced STIRR [18], 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 having two partitions, the user may hesitate in using it due to the complicated and not intuitive working mechanism.

CACTUS [17] 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.

Cheng et al. [13] applied the entropy concept in numerical subspace clustering, and Coolcat [6] introduced the entropy concept into categorical clustering. Coolcat is kind of similar to KModes. However, Coolcat assigns the item to a cluster that minimizes the expected entropy. Considering the cluster centers may shift, 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, it did not consider the problem of finding the optimal number of categorical clusters. Some closely related work also borrows concepts from information theory, including Co-clustering [15], Information Bottleneck [28] and LIMBO [3].

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 only the definition of similarity function for categorical data. Thus, it has the similar problem we described.

Most of the recent research in categorical clustering is focused on clustering algorithms. Surprisingly, there is little research concerning about the cluster validation problems for categorical datasets.

7 Conclusion

Most of the recent research about categorical clustering has only contributed to categorical clustering algorithms. In this paper, we proposed an entropy-based cluster validation method for identifying the best Ks for categorical data clustering. Our method suggests to find the best Ks by observing the "Entropy Characteristic Graph (ECG)", which describes the entropy property of partitions with varying K and is significant in characterizing the clustering structure of categorical data. The "Best-K plot (BkPlot)" is used to find the significant points conveniently from the Entropy Characteristic Graph. BkPlots generated by different algorithm may have different performance in iden-

tify the significant clustering structures. In order to find the robust BkPlot, We also develop an entropy-based agglomerative hierarchical algorithm HierEntro. Our experiments show that, HierEntro can generate the most robust BkPlots for various experimental datasets, compared to the other two entropy-based algorithms: Monte-Carlo algorithm and Coolcat algorithm. Meanwhile, HierEntro can also find high quality clustering results in terms of the entropy criterion. Therefore, BkPlot validation method together with HierEntro algorithm can serve as an effective tool for analyzing the significant clustering structures of categorical datasets.

8 Acknowledgement

This research is partially supported by NSF CNS, NSF CCR, NSF ITR, DoE SciDAC, DARPA, CERCS Research Grant, IBM Faculty Award, IBM SUR grant, HP Equipment Grant, and LLNL LDRD.

References

[1] C. C. Aggarwal, C. Magdalena, and P. S. Yu. Finding localized associations in market basket data. *IEEE Trans. on Knowledge and Data Eng.*, 14(1):51–62, 2002.

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

[3] P. Andritsos, P. Tsaparas, R. J. Miller, and K. C. Sevcik. Limbo:scalable clustering of categorical data. *Proc. of Intl. Conf. on Extending Database Technology (EDBT)*, 2004.

[4] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. *Proc. of ACM SIGMOD Conference*, 1999.

[5] D. Barbara and S. Jajodia, editors. *Applications of Data Mining in Computer Security*. Klumer Academic Publishers, 2002.

[6] D. Barbara, Y. Li, and J. Couto. Coolcat: an entropy-based algorithm for categorical clustering. *Proc. of ACM Conf. on Information and Knowledge Mgt. (CIKM)*, 2002.

[7] F. Baulieu. Two variant axiom systems for presence/absence based dissimilarity coefficients. *Journal of Classification*, 14, 1997.

[8] A. Baxevanis and F. Ouellette, editors. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, 2nd edition*. Wiley-Interscience, 2001.

[9] H. Bock. Probabilistic aspects in cluster analysis. *Conceptual and Numerical Analysis of Data*, 1989.

[10] M. Brand. An entropic estimator for structure discovery. In *Proc. Of Neural Information Processing Systems (NIPS)*, pages 723–729, 1998.

[11] G. Celeux and G. Govaert. Clustering criteria for discrete data and latent class models. *Journal of Classification*, 1991.

[12] K. Chen and L. Liu. Vista: Validating and refining clusters via visualization. *Information Visualization*, 3(4), 2004.

[13] C. H. Cheng, A. W.-C. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. *Proc. of ACM SIGKDD Conference*, 1999.

[14] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.

[15] I. S. Dhillon, S. Mellela, and D. S. Modha. Information-theoretic co-clustering. *Proc. of ACM SIGKDD Conference*, 2003.

[16] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Second International Conference on Knowledge Discovery and Data Mining*, 1996.

[17] V. Ganti, J. Gehrke, and R. Ramakrishnan. CACTUS-clustering categorical data using summaries. *Proc. of ACM SIGKDD Conference*, 1999.

[18] D. Gibson, J. Kleinberg, and P. Raghavan. Clustering categorical data: An approach based on dynamical systems. *Proc. of Very Large Databases Conference (VLDB)*, 8(3–4):222–236, 2000.

[19] S. Guha, R. Rastogi, and K. Shim. ROCK: A robust clustering algorithm for categorical attributes. *Proc. of IEEE Intl. Conf. on Data Eng. (ICDE)*, 1999.

[20] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Cluster validity methods: Part I and II. *SIGMOD Record*, 31(2):40–45, 2002.

[21] Z. Huang. 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.

[22] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice hall, 1988.

[23] A. K. Jain and R. C. Dubes. Data clustering: A review. *ACM Computing Surveys*, 31, 1999.

[24] E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer-Verlag, 1998.

[25] T. Li, S. Ma, and M. Ogihara. Entropy-based criterion in categorical clustering. *Proc. of Intl. Conf. on Machine Learning (ICML)*, 2004.

[26] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[27] S. Sharma. *Applied Multivariate Techniques*. Wiley&Sons, 1995.

[28] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 1999.

[29] N. Wrigley. *Categorical Data Analysis for Geographers and Environmental Scientists*. Longman, 1985.

[30] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55:311–331, 2004.

Appendix:Proof of Proposition 1 We can expand Proposition 1 with the definition of entropy in section 2.

$$-\sum_{j=1}^{d}\sum_{v\in A_j}(n_p+n_q)p(x_j=v|C_p\cup C_q)\cdot \log_2 p(x_j=v|C_p\cup C_q) \geqslant$$
$$-\sum_{j=1}^{d}\sum_{v\in A_j} n_p p(x_j=v|C_p)\log_2 p(x_j=v|C_p) -$$
$$-\sum_{j=1}^{d}\sum_{v\in A_j} n_q p(x_j=v|C_q)\log_2 p(x_j=v|C_q) \quad (2)$$

It is equivalent to check if the following relation is satisfied for each value v in each $domain(A_j)$.

$$n_p p(x_j=v|C_p)\log_2 p(x_j=v|C_p) + n_q p(x_j=v|C_q)\log_2 p(x_j=v|C_q) \geqslant (n_p+n_q)p(x_j=v|C_p\cup C_q)\cdot \log_2 p(x_j=v|C_p\cup C_q) \quad (3)$$

Without loss of generality, suppose C_p having x items and C_q having y items in value v at j-th attribute. The formula 3 can be transformed to $x\log_2\frac{x}{n_p}+y\log_2\frac{y}{n_q}\geqslant(x+y)\log_2\frac{x+y}{n_p+n_q}$. Since x,y,n_p,n_q are positive integers, let $x=s\cdot y$ and $n_p=r\cdot n_q,(s,r>0)$, and then we can eliminate $\log_2$ to get a simpler form: $\frac{r^s}{(1+r)^{s+1}}\leqslant\frac{s^s}{(1+s)^{1+s}}$. It is easy to prove that $\frac{s^s}{(1+s)^{1+s}}$ is the maximum value of the function $f(r)=\frac{r^s}{(1+r)^{s+1}}(r,s>0)$. Therefore, formula (3) is true, thus (2) is true and Proposition 1 is proved. □