Under consideration for publication in Knowledge and Information Systems

"Best K": Critical Clustering Structures in Categorical Datasets

Keke Chen[1], Ling Liu[2]

[1]Department of Computer Science and Engineering, Wright State University, Dayton OH, USA;
[2]College of Computing Georgia Institute of Technology, Atlanta GA, USA

Abstract. The demand on cluster analysis for categorical data continues to grow over the last decade. A well-known problem in categorical clustering is to determine the best K number of clusters. Although several categorical clustering algorithms have been developed, surprisingly, none has satisfactorily addressed the problem of Best K for categorical clustering. Since categorical data does not have an inherent distance function as the similarity measure, traditional cluster validation techniques based on geometric shapes and density distributions are not appropriate for categorical data. In this paper, we study the entropy property between the clustering results of categorical data with different K number of clusters, and propose the *BKPlot* method to address the three important cluster validation problems: 1) How can we determine whether there is significant clustering structure in a categorical dataset? 2) If there is significant clustering structure, what is the set of candidate "best Ks"? 3) If the dataset is large, how can we efficiently and reliably determine the best Ks?

Keywords: Categorical Data Clustering; Entropy; Cluster Validation

1. Introduction

Data clustering is well-known as an important tool in data analysis, where a clustering algorithm uses some similarity measure to group the most similar items into clusters (Jain and Dubes, 1999). Clustering techniques for categorical data are very different from those for numerical data in terms of the definition of similarity measure. Traditionally, categorical data clustering is merged into numerical clustering through a data preprocessing stage (Jain and Dubes, 1999). In the preprocessing, numerical features are extracted/constructed from the categorical data, or a conceptual similarity function between data records is defined based on the domain knowledge.

Received May 13, 2007
Revised Jun 16, 2008
Accepted Jul 13, 2008

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 directly clustering the raw categorical data is important for many applications. Examples include environmental data analysis (Wrigley, 1985), market basket data analysis (Aggarwal, Magdalena and Yu, 2002), DNA or protein sequence analysis (Baxevanis and Ouellette, 2001), text mining (Wang and Karypis, 2006), and security (Barbara and Jajodia, 2002). Therefore, recently there are increasing interests in clustering categorical data (Huang, 1997; Guha, Rastogi and Shim, 2000; Ganti, Gehrke and Ramakrishnan, 1999; Gibson, Kleinberg and Raghavan, 2000; Barbara, Li and Couto, 2002; Dhillon, Mellela and Modha, 2003; Andritsos, Tsaparas, Miller and Sevcik, 2004; Li, Ma and Ogihara, 2004; Yu, Qian, Lu and Zhou, 2006).

Cluster Validation for Categorical Data Different clustering algorithms hardly generate the same clustering result for the same dataset, and we need cluster validation methods to evaluate the quality of clustering results (Sharma, 1995; Jain and Dubes, 1988; Halkidi, Batistakis and Vazirgiannis, 2002). Formally, there are three main issues in cluster validation: 1) how to evaluate the quality of different partition schemes generated by different clustering algorithms for the same dataset, with a fixed K number of clusters; 2) how to determine whether there is significant clustering structure in the datasets; 3) how to determine the best number of clusters (the "best K"), if there is inherent significant clustering structure in the dataset.

In addition, when large datasets are processed for clustering, the performance becomes critical to effective to both clustering and cluster validation. General approaches addressing the performance issue, such as sampling, raise particular problems for validating clusters in large datasets: 1) whether the result on sample datasets is consistent with that on the original dataset, and 2) how to estimate the proper sample size that guarantees the consistency.

For numerical data, the clustering structure is usually validated by the geometry and density distribution of clusters. When a distance function is given for the numerical data, it is natural to introduce the density-based methods (Ester, Kriegel, Sander and Xu, 1996; Ankerst, Breunig, Kriegel and Sander, 1999) into clustering. As a result, the distance functions and density concepts play unique roles in validating the numerical clustering result. Various statistical cluster validation methods and visualization-based validation methods have been proposed for numerical data (Jain and Dubes, 1988; Halkidi et al., 2002; Chen and Liu, 2004), all of which are based on the geometry and density property of datasets. 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.

Due to the lack of intuitive distance functions between categorical values, the techniques used in cluster validation for numerical data are not applicable to categorical data. Without reasonable numerical feature extraction/construction for a given categorical dataset, the general distance functions are usually inapplicable. As a result, no geometry/density-based validation method is appropriate in validating the clustering result for categorical data. Surprisingly, to our knowledge, there is no literature satisfactorily addressing the cluster validation problems for categorical data.

Entropy Based Categorical Clustering One way to address the similarity problem for categorical data is to use set-based similarity measures, for example, the *entropy* (Cover and Thomas, 1991) based measures. Originated from information theory, en-

tropy has been applied in both pattern discovery (Brand, 1998) and numerical clustering (Cheng, Fu and Zhang, 1999). Recently, there have been some efforts in applying entropy and the related concepts in information theory to clustering categorical data (Barbara et al., 2002; Li et al., 2004; Dhillon et al., 2003; Andritsos et al., 2004; Chen and Liu, 2005). Initial results have shown that entropy criterion can be very effective in clustering categorical data.

Entropy-based similarity measures evaluate the orderliness of a given cluster. In entropy-based categorical clustering, the quality of clustering result is naturally evaluated by the entropy of all clusters (Barbara et al., 2002; Li et al., 2004), namely, the *expected entropy*. The lower the expected entropy is, the more ordered the clusters are. While it is intuitive to evaluate the overall orderliness of a clustering result with expected entropy, can entropy also be used to identify the best K number of clusters? can it be used to determine whether there is significant clustering structure in a given dataset?

Our Contributions We try to answer the above cluster validation problems based on the entropy difference between the optimal clustering structures. Intuitively, if the best clustering structure has K clusters, fitting the data from K clusters into $K-1$ clusters will seriously disturb the clustering structure, while the change of optimal $K+1$ clusters to K clusters should be much less distinctive. This heuristic leads us to explore the property of *optimal neighboring clustering structures* with K and $K+1$ clusters, respectively, K varying from one to a small number, e.g., $K < 20$, for primary clustering structures, i.e., those consisting of major clusters [1]. The optimality is evaluated based on expected entropy. Briefly, we identify and interpret that the *similarity between the optimal clustering structures* is the key to find the critical clustering structures, and then propose the "*Best-K Plot* (BKPlot)" method to conveniently capture the dramatic difference between the optimal clustering structures.

However, optimal BKPlots are based on optimal clustering results, which are computationally intractable due to the NP-hard complexity of entropy minimization. Generating high-quality "approximate BKPlots" becomes a significant problem in practice. We address this problem with two aspects. First, we propose a new inter-cluster similarity measure *Incremental Entropy (IE)*. With IE, a standard agglomerative categorical clustering algorithm with Entropy criterion (as "ACE algorithm referred in this paper) can be used for generating reliable approximate BKPlots. The initial experimental results show that an agglomerative algorithm based on IE can generate high-quality BKPlots, compared to other entropy-criterion based algorithms. Second, for large datasets, we develop the theory of *sample approximate BKPlot*, which addresses the issues in consistently identifying the Best Ks based on uniformly sampled datasets with the IE-based method.

When applying sample approximate BKPlot, we also notice that there are datasets having no significant clustering structure, such as uniformly distributed data, or single-mode clustering structure. By exploring the characteristics of the BKPlots for the typical no-cluster datasets, we provide a testing method for determining whether a given dataset has significant clustering structure.

In summary, we propose a framework for determining the critical clustering structure in categorical datasets, which consists of four main components

1. the basic BKPlot method for determining the best Ks;

[1] Major clusters are usually large clusters in terms of the overall size of the dataset. Small clusters will become a part of some large cluster in a hierarchical clustering structure, or be regarded as outliers. Our technique does not intend to identify clustering structures that distinguishes small clusters.

2. the incremental entropy based inter-cluster similarity measure and the algorithm for generating reliable approximate BKPlots;
3. the theory of sample approximate BKPlot for large datasets;
4. a testing method for determining whether there is significant clustering structure for a given dataset.

The rest of the paper is organized as follows. Section 2 sets down the basic concepts and notations. Section 3 introduces the BKPlot method for determining the best Ks and the intuition behind the method, as well as an effective method for generating approximate BKPlots - the ACE algorithm. In section 4 and 5, we analyze the sample approximate BKPlots for large datasets, and also propose a novel testing method for determining the existence of significant clustering structure. The experimental results are presented in Section 6. Section 7 reviews some related work in categorical clustering and cluster validation, and finally we conclude the paper.

2. Basic Notations and Concepts

We define the notations used in this paper and then introduce the entropy-based clustering criterion. Some basic properties about the entropy criterion will be presented in the later sections.

2.1. Basic Entropy Definition

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. 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 introduce the classical entropy definition (Cover and Thomas, 1991).

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

Here, entropy $H(X)$ is based on the column entropy $H(x_j)$, while the correlations between columns are ignored for easy manipulation without affecting the results (Barbara et al., 2002; Li et al., 2004). $H(X)$ is often estimated with the sample set $\mathbb{S}$, we define the estimated entropy as $H(X \mid \mathbb{S})$.

$$H(X \mid \mathbb{S}) = -\sum_{j=1}^{d} \sum_{v \in A_j} p(v \mid \mathbb{S}) \log_2 p(v \mid \mathbb{S})$$

where $p(v \mid \mathbb{S})$ is the empirical probability estimated on $\mathbb{S}$. To further simplify the notation, we also define the **column entropy** of A_j as

$$H(A_j \mid \mathbb{S}) = -\sum_{v \in A_j} p(v \mid \mathbb{S}) \log_2 p(v \mid \mathbb{S})$$

The estimated entropy on $\mathbb{S}$ is simply the sum of the column entropies.

Now we can define the concept of cluster entropy. Suppose the dataset $\mathbb{S}$ is partitioned into K clusters. Let $C^K = \{C_1, \ldots, C_K\}$ represent the partition, where C_k is a cluster and n_k represent the number of records in C_k. Thus, the **cluster entropy** of C_k is the dataset entropy $H(X \mid C_k)$. For simpler presentation, we use $H(\mathbb{S})$ and $H(C_k)$ to represent the dataset entropy and cluster entropy, respectively.

The classical entropy-based clustering criterion tries to find the optimal partition, C^K, which maximizes the following entropy criterion (Bock, 1989; Celeux and Govaert, 1991; Li et al., 2004).

$$Opt(C^K) = \frac{1}{d}\left(H(\mathbb{S}) - \frac{1}{n}\sum_{k=1}^{K} n_k H(C_k)\right)$$

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

Li et al (Li et al., 2004) showed that the minimization of expected-entropy can be unified into probabilistic clustering framework, and closely related to many important concepts in information theory, clustering and classification, such as Kullback-Leibler Measure (Cover and Thomas, 1991), Maximum Likelihood (Lehmann and Casella, 1998), Minimum Description Length (Cover and Thomas, 1991), and dissimilarity coefficients (Baulieu, 1997). 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 (Cheng et al., 1999), it is not the best choice for capturing all of the geometric properties a numerical dataset may have.

2.2. Incremental Entropy

Individually, cluster entropy cannot determine the structural difference between clusters. However, we observe that the structural difference can be observed by mixing (merging) two clusters. By entropy definition, the structural characteristic of a dataset is determined by the value frequencies in each column. Intuitively, mixing two clusters that are similar in the inherent structure will not change the value frequencies, thus, not change the expected-entropy of the partition as well. However, merging dissimilar ones will inevitably change the value frequencies, increasing the expected-entropy. Therefore, the increase of expected entropy in merging clusters has some correlation with the similarity between clusters.

By the definition of expected-entropy, after merging two clusters in a partition the difference of expected-entropy can be equivalently evaluated by the difference between the weighted entropies, i.e., $(n_p + n_q)H(C_p \cup C_q)$ and $n_p H(C_p) + n_q H(C_q)$. We have the following first result about weighted entropies.

Proposition 2.1. $(n_p + n_q)H(C_p \cup C_q) \geqslant n_p H(C_p) + n_q H(C_q)$

PROOF SKETCH. This proposition formally states that mixing two clusters will not reduce the weighted entropy. The first step of the proof is to expand both sides of the formula with the entropy definition. Let $p(x_j = v|C_p)$ be the estimated probability of $x_j = v$ in the column A_j within the cluster C_p.

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

It is straightforward to prove that the above formula is true if the following relation is satisfied for each value v in each column A_j. Namely, if we can prove that for each categorical value in each column the following formula is true, then the proposition is established.

$$n_p p(x_j=v|C_p)\log_2 p(x_j=v|C_p)+n_q p(x_j=v|C_q)\log_2 p(x_j=v|C_q)$$
$$\geqslant (n_p+n_q)p(x_j=v|C_p\cup C_q)\cdot\log_2 p(x_j=v|C_p\cup C_q) \quad (2)$$

Without loss of generality, suppose C_p having x rows and C_q having y rows with value v at j-th attribute, $x,y>0$ (if $x=0$ or $y=0$, the inequality is trivially satisfied), i.e., $p(x_j=v|C_p)=\frac{x}{n_p}$, $p(x_j=v|C_q)=\frac{x}{n_q}$,and $p(x_j=v|C_p\cup C_q)=\frac{x+y}{n_p+n_q}$. Then, the inequality 2 can be transformed to $x\log_2\frac{x}{n_p}+y\log_2\frac{y}{n_q}\geqslant(x+y)\log_2\frac{x+y}{n_p+n_q}$, which is exactly the "log-sum inequality" (Cover and Thomas, 1991). □

We define the key concept "*Incremental Entropy* (IE)" based on the above proposition.

Definition 2.1. Incremental Entropy reflects the structural difference between two clusters, which is quantified by $IE(C_p,C_q)=(n_p+n_q)H(C_p\cup C_q)-(n_pH(C_p)+n_qH(C_q))$.

From the proof of Proposition 2.1, it follows that if the two clusters have the *identical structure*, i.e., for every categorical value v_i in every attribute x_j, $1\leqslant i\leqslant|A_j|$, $1\leqslant j\leqslant d$, $p(v_i|C_p)=p(v_i|C_q)$ is satisfied, $IE(C_p,C_q)=0$. Interestingly, identical structure does not concern the size of the clusters. Thus, it intuitively implies that sampling will be effective when IE is used as a clustering criterion, which we will use to derive another important result later.

Incremental entropy is a critical measure in our method. We will use this similarity measure to construct a hierarchical clustering algorithm in order to generate reliable results for approximately determining the best Ks. The relationship between cluster merging and similarity can also help us understand the method to find the best Ks in next section.

For clear presentation, we summarize the notations we will use in Table 1.

3. Finding the Best K with Entropy Criterion

Traditionally, the Best Ks for numerical data clustering are identified with statistical index curves, based on geometry and density properties of the dataset (Sharma, 1995; Halkidi et al., 2002) or likelihood (Hastie, Tibshirani and Friedmann, 2001). Depending

notation	description
d, N, n	d: the number of attributes, N: the size of dataset, n: the sample size
A_j, $\|A_j\|$	represents attribute j, and $\|A_j\|$ is the number of distinct categorical values in this attribute
$H(C_k)$	entropy of cluster C_k
$H(A_j \mid C_k)$	column entropy of column A_j in cluster C_k
$IE(C_p, C_q)$	incremental entropy between clusters C_p and C_q
$\bar{H}(C^K)$	expected entropy of a K-cluster partition
$\bar{H}_{opt}(C^K)$	optimal (minimum) expected entropy of all K-cluster partitions
$I(K)$	entropy difference of a pair of optimal neighboring partitions, i.e., $\bar{H}(C^K) - \bar{H}(C^{K+1})$
$B(K)$	the BKPlot function, derived from I(K)

Table 1. Notations.

on the different property of the index curve, the Ks at peaks, valleys, or distinguished "knees" on the curve, may be regarded as the candidates of the optimal number of clusters (the best Ks). *Are there entropy-based such curves indicating the significant clustering structures for categorical data?*

The first thought might be investigating the curve of expected entropy for the optimal partitions. We define

Definition 3.1. An optimal partition of K clusters is a partition that results in the minimum expected entropy among all K-cluster partitions.

The expected entropy of the optimal partition is denoted by $\bar{H}_{opt}(C^K)$. Our result shows that the $\bar{H}_{opt}(C^K)$ curve is often a smoothly decreasing curve without distinguished peaks, valley, or knees (Figure 1). However, we can actually obtain some information between the neighboring optimal partitions (with K and $K+1$ clusters respectively) with the concept of entropy difference. Concretely, the difference of neighboring expected-entropies (Figure 2) can be used to indicate the critical clustering structures. This relationship can be intuitively illustrated and understood by merging similar clusters in an optimal partition. The entropy-difference curve often shows that the similar partitions 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 of the best Ks. Before going to details, we first give some entropy properties of optimal partitions.

3.1. Entropy Properties of Optimal Partitions

Given the number of clusters, K, there is at least one optimal partition with minimum expected entropy $\bar{H}_{opt}(C^K)$. There are several properties about $\bar{H}_{opt}(C^K)$.

First of all, $\bar{H}_{opt}(C^K)$ is bounded. It is easy to see that $\bar{H}_{opt}(C^K)$ is less than the dataset entropy $H(\mathbb{S})$. $\bar{H}_{opt}(C^K)$ is maximized when $K = 1$ – all data points are in the same cluster. We also have $\bar{H}_{opt}(C^K) \geqslant 0$ as the entropy definition implies. The zero entropy $\bar{H}_{opt}(C^k)$ is reached at $k = N$, when each record is a cluster. Therefore, $\bar{H}_{opt}(C^K)$ is bounded by $[0, H(\mathbb{S})]$.

We can also derive the relationship between the optimal partitions with any different number of clusters, K and L, $K < L$, with the help of Proposition 2.1.

Proposition 3.1. $\bar{H}_{opt}(C^K) \geqslant \bar{H}_{opt}(C^L)$, when $K < L$

PROOF SKETCH. Let a L-cluster partition C_0^L be formed by splitting the clusters in the

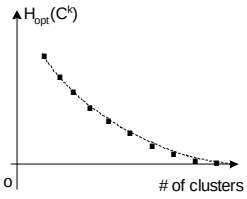

Fig. 1. Sketch of expected entropy curve.

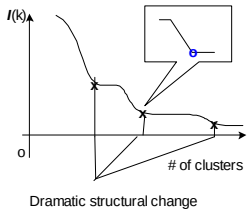

Fig. 2. Sketch of entropy-difference curve of neighboring partitions.

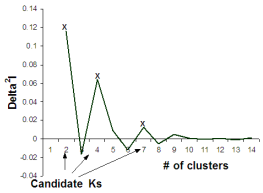

Fig. 3. Finding the best k with BKPlot (example of soybean-small data).

optimal K-cluster partition. With Proposition 2.1 and the definition of optimal partition, we have

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

□

Proposition 3.1 shows that the optimal expected-entropy decreases with the increasing of K, which meets the intuition well. However, it is hard to describe the curve with a function of closed form in terms of K. As our experimental result shows, it is often a negative logarithm-like curve (Figure 1). This curve implies that, 1) it is highly possible that the best K is not unique in terms of entropy criterion, and 2) with only expected-entropy curve, we cannot clearly identify the significant clustering structures.

3.2. Understanding the Similarity of Neighboring Partitions

In this section, we focus on the similarity between the neighboring optimal partitions. We formally define this similarity with the entropy difference between the neighboring partitions as

$$I(K) = \bar{H}_{opt}(C^K) - \bar{H}_{opt}(C^{K+1})$$

There are two heuristics to capture this similarity. One aspect is the absolute value of $I(K)$, 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 not easy to understand the change between the optimal partitions, we use a cluster merging scheme, which will be described in Section 3.3, to demonstrate the two aspects of similarity.

- First, small $I(K)$ means high similarity between the neighboring partitions. We can understand this by merging the similar clusters in K+1-cluster partition to form K-cluster partition. Merging identical clusters introduces zero increase of entropy and the clustering structure is not changed at all. Similarly, small increasing rate between two neighboring schemes implies that the reduction of number of clusters does not introduce significant change to the clustering structure.
- For large change of expected entropy, we first consider the meaning of large $I(K)$. If the expected-entropy increases a lot from K+1 to K, this reduction of number of clusters should introduce considerable impurity into the clusters and thus the clustering structure can be changed significantly. However, whether this change is globally distinguishable from others depends on the further comparison between the continuous

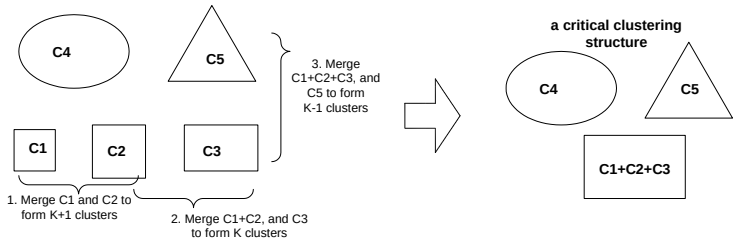

Fig. 4. Similar merges with $I(K) \approx I(K+1)$, but $I(K-1) \gg I(K)$

changes. Consider $I(K)$ as the amount of extra impurity introduced from $K+1$-cluster partition to K-cluster partition. If $I(K) \approx I(K+1)$, i.e. K-cluster partition introduces similar amount of impurity as K+1-cluster scheme does, we consider that the clustering structure is *similarly* changed from K+1-cluster scheme to K-cluster scheme.

An example of "similar merges" in Figure 4 can well demonstrate similar changes vs. significant changes of clustering structure. We use icons to conceptually represent categorical clusters. The shape and the size of an icon represent the structure and the size of the cluster, respectively. Suppose that the initial state with $C1 - C5$ are the optimal partition of K+2 clusters, among which C1, C2, and C3 are in a similar clustering structure as shown in Figure 4. Now, we try to form sub-optimal partitions by merging the similar clusters. As Proposition 2.1 shows, merging the clusters with similar clustering structure will result in small IE, which means small entropy difference between the two neighboring partitions. Suppose that, $H(C^{K+1})$ is approximately minimized by merging C1 and C2, and $H(C^K)$ by merging C1+C2 and C3. Since the three clusters are in a similar structure, the consecutive two merge operations result in similar $I(K)$ and $I(K+1)$. The resultant clustering structures should not be distinguishable from each other. On the other hand, reducing the number of clusters further from K to $K-1$ will change the clustering structure a lot and inevitably bring more impurity into the partitions. As a result, $I(K-1)$ will be much larger than $I(K)$ and $I(K+1)$. We can see that K becomes a significant point where a series of similar clustering structures are changed to a significantly different clustering structure. Therefore, the "knees" on the $I(K)$ curve can provide substantial information about similar changes or significant changes of clustering structure.

In practice, if a dataset has significant clustering structure, we can find a series of neighboring "stable" schemes, which result in similar $I(K)$, and we may also find the critical points where a series of "stable" schemes become "less stable" when K is reduced (Figure 2). All of the critical points should be the candidates of the best Ks and could be interesting to cluster analysis.

The common way to mathematically identify such critical knees on the $I(K)$ curve is to find the peaks/valleys of the second-order difference of the curve. Specifically, since $I(K)$ curve consists of a set of discrete points, we define the second-order difference of the curve as

$$B(K) = \delta^2 I(K) = \delta I(K-1) - \delta I(K)$$

and $\delta I(K) = I(K) - I(K+1)$ to make K aligned with the critical points. For convenience, we name the B curve as the "Best-k Plot (BKPlot)" (Figure 3) and $B(K)$ indicates the value of B function at K. Notice that dramatic structure changes happen only at $I(K) > I(K+1)$. $I(K) < I(K+1)$ means the structure change is slowing

down when the number of clusters is reduced from $K+1$ to K. In this case, we should continue to look at the range $2 \leq k < K$ to find other dramatic changes [2]. Therefore, we need only to look at the peaks of BKPlot to find the best Ks.

3.3. Generating High-quality Approximate BKPlot

From the definition of BKPlot, we know it is intractable to generate the exact BKPlot since obtaining the optimal clustering result is computationally intractable for even a small dataset. However, it is important to note that the use of BKPlots is to find the best Ks, and "approximate BKPLots" may be used to correctly identify the Best Ks as well. We consider high quality BKPlots as the approximate BKPlots that can consistently and correctly identify the candidate best Ks.

Although there are many clustering algorithms proposed so far, none have designed with the above objectives in mind. Hence, we develop a method for generating high-quality approximate BKPlots. Since it is a standard Agglomerative Categorical clustering algorithm with Entropy criterion, we call it ACE for short.

ACE is based on the proposed inter-cluster similarity measure: the Incremental Entropy. While the traditional hierarchical algorithms for numerical clustering need to explicitly define the inter-cluster similarity with "single-link", "multi-link" or "complete-link" methods (Jain and Dubes, 1988), incremental entropy is a natural cluster-based similarity measure, ready for constructing a hierarchical clustering algorithm.

As a standard hierarchical algorithm, ACE 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. the incremental entropy $IE(C_p, C_q)$ is minimum among all pairs of clusters.

Since IE calculation involves the expensive entropy calculation, a working algorithm has to optimize the entropy calculation. ACE uses three structure to maintain the incremental calculation of IE values: *summary table* for convenient counting of occurrences of values, IE*-table* for bookkeeping $IE(C_p, C_q)$ of any pair of clusters C_p and C_q, and a IE *heap* for maintaining the minimum IE value in each merge. With the help of these data structures, we can optimize the IE based ACE algorithm (with complexity of $O(N^2 logN)$). We will simply skip the detailed algorithm here, but refer interested readers to the paper (Chen and Liu, 2005). Experimental results show that ACE algorithm is the most effective algorithm for generating high-quality approximate BKPlots, compared to other existing algorithms similarly optimizing entropy criterion, such as Monte-Carlo (Li et al., 2004), Coolcat (Barbara et al., 2002), and LIMBO (Andritsos et al., 2004).

More importantly, the ACE algorithm has a nice property in generating sample BKPlots for handling large datasets − The mean of sample BKPlots generated by ACE algorithm on sample datasets is essentially an unbiased estimator of the original BKPlot generated by ACE for the entire dataset. This property is important since we often need to handle large datasets that cannot be directly processed by the ACE algorithm.

[2] The nature of dramatic entropy change also implies that the identified clustering structures are major clustering structures, which consist of large clusters in terms of the size of dataset. Note that small clusters will be under the "shadow" of these large clusters. An effective methods to explore small clusters will be iteratively focusing on a part of the dataset based on the identified structure.

notation	description
$\hat{H}(C_{n,k})$	estimated entropy of cluster C_k with sample size n
$I(n,K)$	approximate I(K) with sample size n
$B(n,K)$	approximate B(K) with sample size n
$E[], Var()$	expectation and variance of approximate B(K) or I(K)

Table 2. Extended Notations for Sample Data.

4. Sample BKPlots for Large Datasets

When the dataset is large (e.g., the number of records $N > 10,000$), ACE algorithm is not effective due to its complexity $O(N^2 \log N)$. There are two commonly accepted approaches to handling large datasets: one is sampling and the other is summarization. We will focus on the uniform sampling approach for finding the best Ks for large datasets in this paper, while the summarization approach has been reported for processing data streams (Chen and Liu, 2006).

Why uniform sampling is also appropriate for BKPlots? The reason is that BKPlot is only used to identify the major change of clustering structure, which consists of large clusters in comparable size [3]. Thus, while uniform sampling is applied with an appropriate sample size, we can assume such clustering structures are well preserved. In the sampling approach, we identify the best Ks based on the BKPlots generated on sample datasets, which we name it *sample BKPlots*, denoted by $B(n,K)$ at sample size n and the number of clusters K. We also name the BKPlot on the original dataset *original BKPlot*. Let s denote the number of sample BKPlots and i denote the sample BKPlot based on the sample dataset i. We define the *mean BKPlot* of s sample BKPlots as follows.

$$E[B(n,K)] = \frac{1}{s}\sum_{i=1}^{s} B_i(n,K)$$

In order to prove that sample BKPlots based on uniform sampling can work effectively for large datasets, we need to answer three questions: 1) Do the mean BKPlots converge to the original BKPlot? 2)How is the quality of mean BKPlots in terms of consistency to the original BKPlot? 3) How to estimate the appropriate sample size that guarantees the mean BKPlot is consistent with the original BKPlot? The following sections will be focused on these three questions.

If mean BKPlots converge to the original BKPlot, we can confidently use mean BKPlots to estimate the original BKPlot. The quality of mean BKPlots should be related to the sample size and possibly other factors. We below first discuss the convergence of mean BKPlots which is used to evaluate the consistency of the mean BKPlots. Then we study the variance of mean BKPlots to evaluate the quality of the BKPlot estimation, and finally we develop the method for verifying whether a given sample size can guarantee a reliable mean BKPlot or not.

4.1. Convergence of mean BKPlots

In general, we model the major clustering structure of a very large dataset $\mathbf{S}$ as follows. Let the primary clustering structure contains a few *large* clusters, denoted as $C^M =$

[3] small clusters will be merged to the "nearby" large clusters and should be explored in secondary structures.

$\{C_1, C_2, \ldots, C_M\}$, e.g., $M = 10$, which are preserved under uniform sampling, with n records. Thus, we can assume the M clusters are proportionally sampled to generate the sample dataset. We denote the sample clusters with $C_n^M = \{C_{n,1}, C_{n,2}, \ldots, C_{n,M}\}$. With the definition of cluster entropy, we have the following proposition.

Proposition 4.1. If the primary clustering structure is preserved with sample size n, the sample cluster entropy $\hat{H}(C_{n,i})$ converges to $H(C_i)$, when $n \to N$

This proposition states that with the increasing sample size, the structure of sample cluster become more and more similar to that of original cluster. This can be easily proved based on the definition of subset entropy (please refer to Appendix for details).

Our ultimate goal is to study the mean and variance of $B(n, K)$, which are related to the entropy difference between the sample clustering structures, denoted as $I(n, K)$, where n is the sample size and K is the number of cluster. Let a point on a sample BKPlot be $B(n, K)$. We show that we can use a set of sample BKPlots to estimate the original BKPlot for the entire large dataset as the following theorem states.

Theorem 1. The original BKPlot generated by the ACE algorithm for the large dataset can be estimated with the mean of sample BKPlots that are also generated by ACE.

PROOF SKETCH. The proof consists of two steps.

1) $E[I(n, K)]$ converges to the approximate $I(K)$ generated by the ACE algorithm. When the ACE algorithm is used to generate BKPlots, the entropy difference $I(K)$ between the nearby clustering schemes is $I(K) \approx \frac{1}{N} IE^{(K)} = \frac{1}{N}\{(N_p+N_q)\hat{H}(C_p+C_q) - N_p\hat{H}(C_p) - N_q\hat{H}(C_q)\}$. Similarly, it applies to the sample datasets, i.e., $I(n, K) \approx \frac{1}{n}\{(n_p + n_q)\hat{H}(C_{n,p} + C_{n,q}) - n_p\hat{H}(C_{n,p}) - n_q\hat{H}(C_{n,q})\}$. Since the mean of sample cluster entropy $E[\hat{H}(C_{n,i})]$ for any cluster i converges to $\hat{H}(C_i)$ due to Proposition 4.1, and $N_p/N \approx n_p/n$, $N_q/N \approx n_q/n$, $E[I(n, K)]$ also converges to the approximate $I(K)$.
2) Since the BKPlot is based on the $I(K)$ curve, i.e., $B(n, K) = I(n, K - 1) - 2I(n, K) - I(n, K + 1)$, the mean of sample BKPlots $E[B(n, K)]$ will also converge to the original BKPlot $B(K)$ generated by the ACE algorithm. □

How reliable $E[B(n, K)]$ can be used to represent the original BKPlot depends on its variance. Before we design the method to check the reliability of a mean BKPlot, we first study the variance of mean BKPlots.

4.2. Variance of Mean BKPlots

The variance of mean BKPlot, $Var(E[B(n, K)])$, can be used to evaluate the quality of BKPlot estimation. We are more interested in the asymptotic variance, especially the relationship between variance and sample size. We will study the asymptotic variance of $E[B(n, K)]$ in four steps. 1) We derive the general form of asymptotic variance of the sum of random variables: $Var(\sum_{i=1}^{m} a_i X_i)$, which will be heavily used in deriving other results; Then, we show that 2) $Var(E[\hat{H}(C_{n,i})]) \sim O(\frac{1}{n_i s})$; 3)$Var(E[I(n, K)]) \sim O(\frac{1}{ns})$; and 4) finally, we have

$$Var(E[B(n, K)]) \sim O(\frac{1}{ns}) \tag{3}$$

Please refer to the Appendix for the details in the four steps. With the final result – the step 4), it is intuitive that, by increasing the sample size, n, or the number of sample sets, s, the mean BKPlot should become more and more reliable.

4.3. Conditions for Reliable Mean BKPlot

In general, the larger n and s are, the smaller the variance of mean BKPlots is, which results in more reliable estimation. However, in practice, we want to use limited sample size (n) and limited number of sample BKPlots (s) to get a reliable mean BKPlot that is consistent with the original BKPlot. In last section, we have found the relationship between the variance and these factors, with which we are able to verify whether a given set of sample datasets can generate a reliable mean BKPlot and how we can improve the reliability.

We first define the concept of "consistent mean BKPlot". Suppose that the top κ number of candidate Ks (e.g., $\kappa = 3$ and $K < 20$) on the original BKPlot have $B(K)$ values *significantly* higher than certain level η. Without loss of generality, let $k_1 < k_2 < \ldots < k_\kappa$ be the significant Ks on the original BKPlot ordered by $B(k_i)$, $1 \leq i \leq \kappa$.

Definition 4.1. If the sequence of significant Ks on the mean BKPlot $k'_1 < k'_2 < \ldots < k'_\kappa$ satisfies $k'_i = k_i$ for $1 \leq i \leq \kappa$, the mean BKPlot is consistent with the original BKPlot.

There are $\kappa + 1$ significant levels on the original BKPlot: $B(k_i)$, $1 \leq i \leq \kappa$, and η. Let the minimum difference between the $\kappa + 1$ values be Δ determined by certain $B(k_r)$ and $B(k_q)$, $r < q$, i.e., $\Delta = |B(k_r) - B(k_q)|$. Let c be the constant related to certain confidence level (Lehmann and Casella, 1998), for example, $c = 1.96$ for $CL = 95\%$ confidence level for normal distribution. Since $E[B(n, K)]$ follows a normal distribution with mean $B(K)$ and variance $Var(B(n, K))/s$ for large n, we can estimate whether the order will be preserved for certain sample size. Concretely, in order to make the κ levels consistent with the original BKPlot, we should guarantee the non-overlapping confidence intervals of $B(n, k'_i)$ at the significant Ks, with confidence CL, i.e., the following constraint is satisfied.

$$c(\sqrt{Var(B(n, k'_i))/s} + \sqrt{Var(B(n, k'_j))/s})$$
$$= 2c\sqrt{Var(B(n, K))/s} < \Delta, 1 \leq i \leq \kappa$$

It follows that, if the sample variance at size n satisfies the following formula, the mean BKPlot will be consistent with the original BKPlot.

$$Var(B(n, K)) < \frac{\Delta^2 s}{4c^2} \tag{4}$$

This describes the reliability of the mean BKPlot in term of sample size n, the number of sample BKPlots s, and the interested top Ks. Note that we really do not know the exact Δ value. We can only bootstrap with some initial n and s for the given dataset, e.g., $n = 1000$ and $S = 10$. Then, we get the estimation of $Var(B(n, K))$ and Δ from the sample BKPlots. If the Formula (4) is satisfied with the estimated $Var(B(n, K))$ and Δ, the mean of sample BKPlots is reliable under certain confidence level CL. Otherwise, we should increase either n, or s, and apparently, increasing s is more economic.

5. Identifying No-cluster Datasets with BKPlot Method

Finding the peaks on BKPlot does not always mean the existence of significant clustering structure. Low peak levels mean smooth changes of clustering structure and each clustering structure does not distinguish itself from others, i.e., there is no significant clustering structure. Therefore, a challenge comes − how to distinguish datasets having clustering structures from those having no clustering structure? We propose a method to address this problem. The main idea is based on the property of the sample BKPlots for the datasets that are known to have *no* clustering structure. Specifically, first, we study the characteristics of clustering structure for the typical no-cluster datasets: datasets with uniform or normal data distribution (single cluster mode). Then, we test if the clustering structure of the given dataset is significantly different from the no-cluster structures.

5.1. Property of Datasets Having No Clustering Structure

We study two types of typical datasets that do not have significant clustering structure. One is the datasets with elements uniformly distributed in a column and between columns, which obviously have no clustering structure. The other type is the datasets of discretized multidimensional normal distribution, which has only one mode (or one cluster). In the following discussion, the Maximum Peak Level (MPL) of BKPlot is used to represent the significance level of the clustering structure.

Ideally, large sample datasets exactly following uniform/normal distribution have no significant clustering structure. Thus, every point on their BKPlots should be very close to zero, with tiny variance for large sample datasets. However, synthetic sample datasets usually do not exactly follow the desired distribution, which may create some small noisy structures. These noisy structures should be treated as statistically equivalent to the ideal no-cluster structures. Therefore, we can also treat any datasets that have clustering structures not significantly different from these noisy structures as the no-cluster datasets. With the varying number of records n, columns d, and column cardinalities $|A_j|$, $1 \leq j \leq d$, synthetic sample datasets may deviate from the desired distribution to different extent, with different levels of noisy structures. We will characterize them both formally and experimentally.

In deriving Formula (3), the asymptotic form of the variance of mean BKPlots, we also have a more detailed form of $Var(E[B(n,K)])$ in terms of the three factors n, d, and $|A_j|$ (see Appendix).

$$Var(E[B(n,K)]) \sim O(\sum_{j=1}^{d} \frac{f(|A_j|)}{ns}) \tag{5}$$

$f(|A_j|)$ represents some unknown function of $|A_j|$. For ideal no-cluster datasets, $B(K) \approx 0$ everywhere. Thus, with the increasing n or d, the maximum peak levels should converge to 0. The converging rate is characterized by the corresponding factors in $Var(E[B(n,K)])$. From above formula we are unable to directly determine the effect of $|A_j|$. However, we will show some experimental results to further study the relationship between the three factors and MPLs.

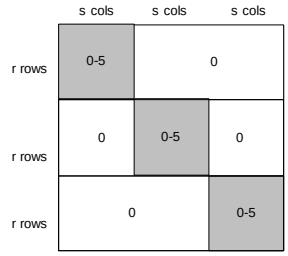

Fig. 5. Synthetic Data DS1

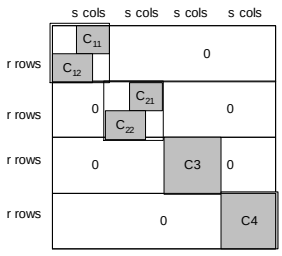

Fig. 6. Synthetic Data DS2

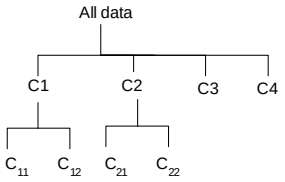

Fig. 7. The significant clustering structures in DS2

5.2. Testing Existence of Significant Clustering Structure

From the above analysis, we conclude that different settings of n, d and $|A|$ may result in different statistical properties of no-cluster BKPlots. There is no simple threshold independent of the settings. Thus, given a real dataset, its sample BKPlots have to be compared to those of no-cluster datasets that have the same setting of n, d, and $|A_j|$.

Combined with the theory of sample BKPlots, we suggest the following testing method for determining the significance of the clustering structure in a given dataset.

1. generate the BKPlot with ACE algorithm for the target dataset and find its MPL, denoted by μ';
2. with the n, d and $|A_j|$ setting of the target dataset, we generate two sets of testing datasets : one with uniform distribution and the other with normal distribution, each with about 30 sample datasets [4];
3. calculate the mean denoted by μ, and the confidence interval of the MPLs of the simulated datasets: $[\mu - CI, \mu + CI]$, at confidence level λ;
4. if $\mu' > \mu + CI$, there is significant clustering structure in the target dataset, otherwise, no clustering structure. CI is often very small compared to the mean level μ, thus, μ is sufficient to represent the upper bound.

6. Experiments

We have formally given the basic BKPlot method, the sample BKPlot method for large datasets, and the BKPlot method for identifying no-cluster datasets. In this section, we want to show that, 1) BKPlots can be used to effectively find the critical Ks; 2) experimental results support the initial analysis of the noisy structures of typical no-cluster datasets; 3) ACE algorithm is a robust tool for generating high-quality approximate BKPlots, compared to the existing entropy-based clustering algorithms, such as Monte-Carlo method (MC) (Li et al., 2004), Coolcat (Barbara et al., 2002), and LIMBO (Andritsos et al., 2004).

[4] a number which is considered as "statistically large"

6.1. Datasets

In order to intuitively evaluate the effectiveness of the method, we use both synthetic and real datasets, and cross-validate the results with the visualization tool VISTA (Chen and Liu, 2004) that is designed for validating numerical clustering results.

Simple Synthetic Data (DS1&DS2). First, we construct two sets of synthetic categorical datasets, so that their clustering structure can be intuitively understood and verified. The first set of datasets has totally separated clusters in one layer 5. The second set has a multi-layered clustering structure with overlapping clusters in the lower layer (the layer with more clusters). Both can be visually verified. Figure 6 shows such a dataset with 1000 records and 30 columns. It has a two-layered clustering structure. The top layer has four clusters, two of which also have two sub-clusters, respectively. 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'. As we can visually identify it, its BKPlot should at least suggest two Best Ks. We name these two types of datasets as DS1 and DS2, respectively.

Discretized Synthetic Mixture Data (DMIX). The third set of datasets, are the discretized version of multidimensional normal mixture datasets (Hastie et al., 2001). We use this set of datasets because we can validate the result with both BKPlot method and the visualization method. As we discretize the continuous value while still preserving the numerical meaning, the original numerical clusters are preserved. Meanwhile, similar values are categorized to the same categorical values, which makes the categorical clustering structure consistent with the numerical clustering structure. We then use a visualization approach, VISTA, which is designed for interactively validating the clustering structure for numerical data (Chen and Liu, 2004), to visually validate the clusters. By doing so, we are able to better understand the nature of our proposed categorical method. A discretized 10-dimensional mixture dataset with 7 clusters (10,000 samples) is visualized in Figure 8. The continuous values in each column are partitioned into 10 equal-width buckets for discretization. These 7 clusters also show a hidden two-layer clustering structure (C1.1, C1.2, C1.3), (C2.1, C2.2), (C3) and (C4). In particular, C2.1 and C2.2 are overlapped by each other, and C1 clusters are very close to each other, which generate some special difficulty for identifying them.

Census Data. We use a real dataset − the discretized version of the large 1990 census data [5]. This dataset is originally used by the paper (Meek, Thiesson and Heckerman, 2002) in studying the relationship between the sampling approach and the effectiveness of Expectation-Maximization (EM) based clustering algorithms for very large datasets. It is large in terms of both the number of records and the number of attributes. After dropping many of the less useful attributes in the raw dataset, the total number of preserved attributes still reaches 68. It contains more than 2 million (2,458,284) records, about 352 megabytes in total. Since the discretized version still preserves the ordinal meaning, we can similarly use the visualization method to evaluate whether the BKPlot method is effective, as we do for DMIX data. C2.1 and C2.2 in Census Data are also a pair of overlapping clusters.

Overlapping clusters are clusters that are similar to each other but each of the clusters still has its own characteristic distribution to distinguish itself from other clusters. Identifying and handling overlapping clusters is a difficult problem in numerical data clustering, and it is more difficult for categorical data. Overlapping clusters in categorical data can be exemplified with these subclusters in DS2, the C2.1 and C2.2 in DMIX,

[5] In UCI KDD Archive http://kdd.ics.uci.edu/databases/census1990/USCensus1990.html

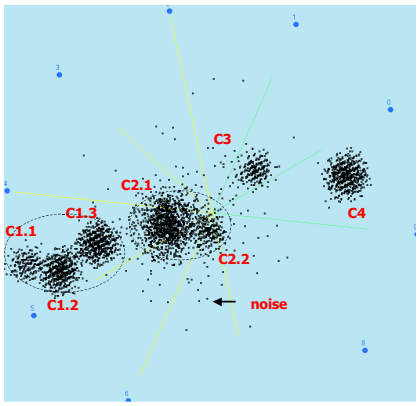

Fig. 8. Visualization of 10K samples of the discretized mixture data DMIX.

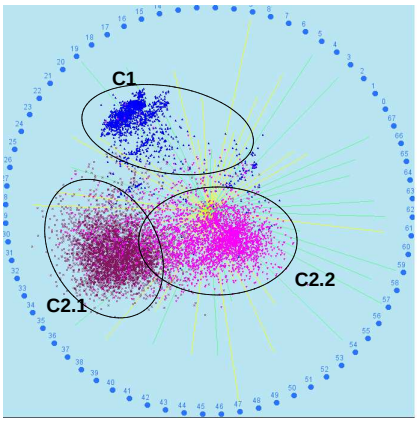

Fig. 9. Sample census data labeled by ACE algorithm.

and also the C2.1 and C2.2 in Census. In many cases, if there are a multi-layered clustering structure, it is possible that overlapping clusters present. Identifying them is useful since deep analysis of these clusters may reveal particular characteristics in practice. Due to the unclear boundary between the overlapping clusters, most methods may fail to identify them. However, we will see that BKPlots can be used to effectively identify them. Further analysis on the grouped clusters identified by the BKPlot method, such as comparing the incremental entropy between each pair of clusters in the group, can reveal more details of the overlapping clusters.

6.2. Validating the BKPlot Method

Below we show the result of applying the BKPlot Method to the synthetic and real datasets. ACE algorithm is used to generate the BKPlots. We generate ten sample datasets for each type of synthetic clustering structure.

DS1. The BKPlots generated by ACE algorithm for DS1-i datasets (Figure 10 clearly indicate that '3' is the only significant K and datasets with the same clustering structure have almost the identical BKPlot. By checking the significant level for the setting of 1,000 records, 30 columns and column cardinality of 6, with 20 no-cluster test datasets (10 normal distribution datasets and 10 uniform distribution datasets, respectively), we find the maximum peak levels (MPLs) of no-cluster datasets are around 0.0004, which is far lower than that of the DS1, 0.21. Therefore, the detected Best K is significant.

DS2. The peaks of BKPlots for DS2-i (Figure 11) include the two inherent significant Ks – '4' and '6'. However, '2' is also given as the third significant K, which suggests that the top 4 clusters can be further clustered into two groups. Interestingly, compared to the three peak levels, we notice that the peak values at 'K=2' have much higher variance, which implies that 'K=2' is less significant than the other two.

DMIX. The BKPlot of DMIX generated by ACE algorithm (Figure 12) indicates that 7 and 4 are the Best Ks with a noisy best K at 2, and they are significant compared to the bound ($\sim$0.0013) for the setting of 1,000 records, 10 columns and column cardinality of 10. The 4-cluster structure indicates that some of the 7 clusters are close to each other and form a secondary clustering structure. The corresponding labeled re-

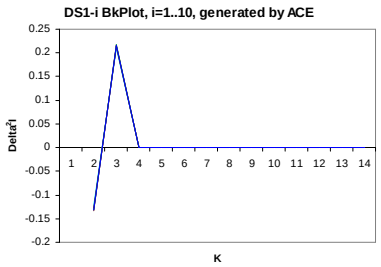

Fig. 10. BKPlot of DS1 by ACE

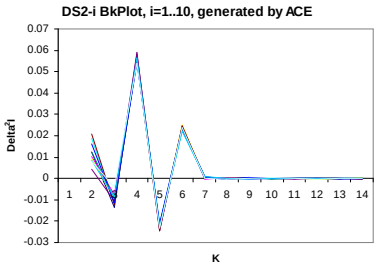

Fig. 11. BKPlot of DS2 by ACE

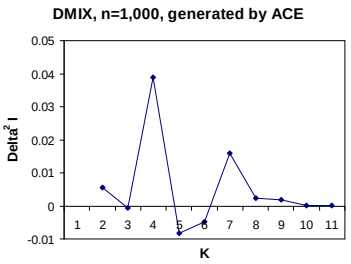

Fig. 12. BKPlot of DMIX by ACE

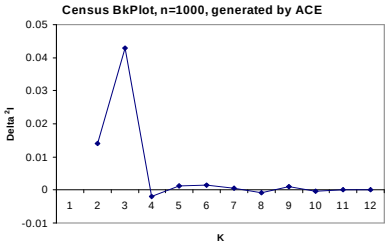

Fig. 13. BKPlot of census dataset by ACE

sult shows the clustering/validation result is well matched with the visualization result (Figure 8).

Census data. The Census data contains three major clusters as shown in the visualization of 1K sample dataset (Figure 9. In the sample data, C2 and C3 are very close, thus, they may also form a two-layer clustering structure − the top layer consists of C1 and C2+C3. Its BKPlot (Figure 13 indicates both 2 and 3 are significant, compared to the bound (~ 0.0005) for the setting (1,000 records, 68 columns and column cardinality defined in (Meek et al., 2002)). The clustering result with ACE at $K = 3$ is very close to the cluster distribution observed via visualization (Figure 9). The cross-validation with the visualization method confirms that the Best K and clustering result generated by ACE algorithm are highly consistent with the inherent clustering structure.

We summarize the detailed information in Table 3. n represents the sample size used in generating BKPlots, d is the number of columns, and "Cardinality" is the column cardinality, i.e., $|A|$. For the first three datasets, each column has the same cardinality. For the census dataset, the column cardinality varies from 2 to 223. Clustering structure describes the possible hierarchical structure of dataset. For example, "two layers, 4/6 clusters" for DS2 means that the clustering structure has two layers with 4 and 6 clusters, respectively. "MPL bounds" is the estimated upper bound of no-cluster datasets with the same setting of n, d and $|A|$. Due to very small confidence intervals, only the mean levels are used to represent the bounds. "BestK" are the best Ks suggested by ACE algorithm and "MPLs at Best Ks" are the corresponding MPLs in the BKPlots generated by ACE algorithm.

Datasets	n	d	cardinality	Clustering Structure	MPL bounds	BestK	MPLs at Best Ks
DS1	1000	30	6	single layer, 3 clusters	0.0004	3	0.21
DS2	1000	30	6	two layers, 4/6 clusters	0.0004	2, 4, 6	0.020, 0.058, 0.023
DMIX	1000	10	10	two layers, 4/7 clusters	0.0013	2,4,7	0.005, 0.041,0.016
Census sample	1000	68	2-223	two layers, 2/3 clusters	0.0005	2,3	0.014,0.044

Table 3. Summary of validating the BKPlot method

6.3. Comparing BKPlot and BIC

Bayesian Information Criterion (BIC) (Hastie et al., 2001) is a popular method for model selection. If appropriate assumption is made about the *prior distribution* of the clusters, it can also be used to select the best K number of clusters. We compare our method with BIC and show the unique advantages of our method.

In order to use BIC method, we need to make assumption about the cluster distribution. Categorical data is often modeled with Multinomial mixture (Lehmann and Casella, 1998). The model fitting is optimized by maximizing the likelihood of fitting the data to the mixture model. The generic form of BIC is then based on the maximum likelihood and the number of parameters used in estimation.

$$BIC = -2 \cdot \log \text{likelihood} + (\log n) \cdot \psi$$

where n is the number of sample records, and ψ is the number of parameters used in the modeling that include the number of clusters. Usually, the K corresponding to the minimum BIC is regarded as the best K. The main problem is, if the real cluster distribution does not well follow the assumed distribution with any K, the result is possibly not good. For instance, in numerical clustering, the Gaussian mixture model assumption does not work well for irregularly shaped clusters and thus BIC based on the Gaussian assumption is not effective.

In experiments, we use AutoClass[6] to generate the fitted model, which will also give BIC values. In AutoClass, ψ is specifically defined as $d(d+1)+K$, where d is the number of columns. On the BIC curve, the best K happens at the minimum BIC. It shows that the BIC method can often suggest one best K, but it cannot find all possible best Ks for multi-layer clustering structures. In the experiment we also observed, the best K for DMIX and Census data is not clearly indicated (Figure 16 and 15), possibly because of the overlapping clusters, the complex clustering structure, and the outliers.

6.4. Properties of Sample BKPlots for Large Datasets

Now we study some properties of sample BKPlots, and particularly mean BKPlots. The mean BKPlot has two important factors: the number s of sample sets and the sample size n for each sample set. We will focus on the size n in the experiment. We use both DMIX and Census data. For each tested sample size, we generate 10 sample BKPlots, on which the mean BKPlot is calculated. Figure 17 for Census data shows, when the clustering structure is simple, the resulting mean BKPlots are very close for different sample sizes, even though the sample size is small. Figure 18 zooms in the peak values

[6] http://ic.arc.nasa.gov/ic/ projects/bayes-group/autoclass/

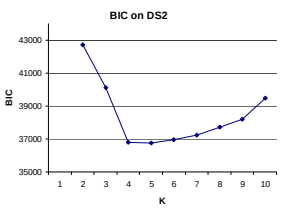

Fig. 14. BIC suggests only K=4 for DS2

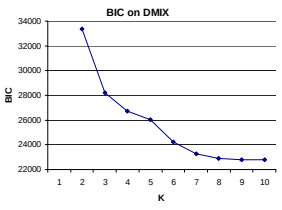

Fig. 15. BIC has no clear suggestion on DMIX data

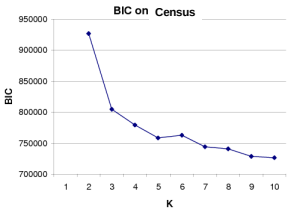

Fig. 16. BIC has no clear suggestion on Census data.

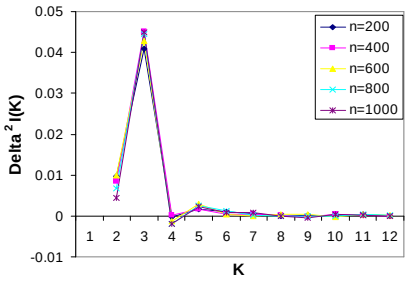

Fig. 17. Comparing the mean BKPlots at different sample size

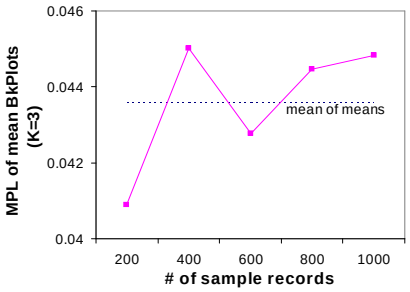

Fig. 18. The most significant values on the mean BKPlots (K=3)

at $K = 3$ at Figure 17. It shows that with the increase of sample size, the MPLs will be stablized at certain level as we expected.

Figure 19 shows that the sample variance decreases with the increasing sample size as we have formally analyzed. For the same sample size, a smaller K also implies merging larger clusters, which usually means a larger $n_p + n_q$. Since Formula (8) in Appendix shows that $var(E[I(n,k)]) \sim O(\frac{n_p+n_q}{n^2 s})$, i.e., the variance is proportional to $n_p + n_q$, a smaller K will result in larger variance.

However, sample size does affect the accuracy of BKPlots for complicated clustering structure. DMIX has more clusters and the clustering structure is also much more

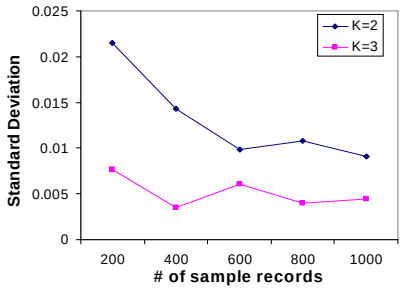

Fig. 19. The variance of the sample BKPlots at K=2 and K=3

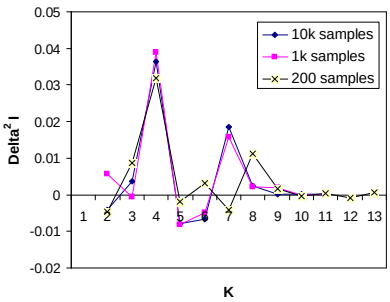

Fig. 20. Mean BKPlots for DMIX at different sample size

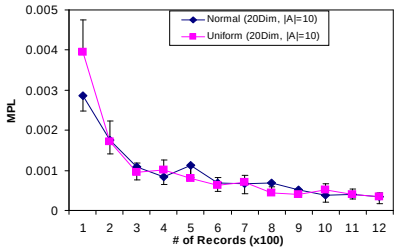

Fig. 21. Relationship between the number of records and MPLs

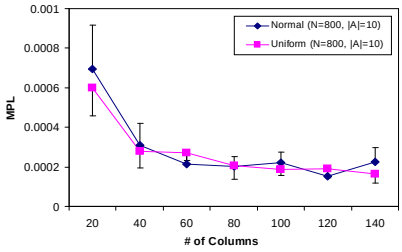

Fig. 22. Relationship between the number of columns and MPLs

complicated. In Figure 20, we see when the sample size is very small, e.g., 200 samples, the clustering structure is not preserved well. The mean BKPlot shows that the low-layer best clustering structure consists of 8 clusters different from the expected 7 clusters. On the contrast, the 4-cluster clustering structure is more stable since it consists of larger number of points. In general, if a cluster is not compact in terms of intra-cluster similarity between points, or small in terms of number of records in the cluster, we will need more samples to preserve it.

6.5. Datasets Having no Clustering Structure

In this set of experiments, we want to show the relationship between the Maximum Peak Levels (MPLs) of no-cluster sample datasets and the factors: the number of records n, the number of columns d, and the column cardinalities $|A_j|$, which should be consistent with our formal analysis.

Since n and d have similar effect on the sample BKPlots according to the analysis, we organize them in the first set of experiments, while the second set of experiments focus on the unknown effect of column cardinality. Each point on the figures is the average of ten runs with the standard deviation as the error bar.

Number of Records and Number of Columns For simplicity, the set of simulated datasets in this experiment have the equal cardinality for columns, denoted as $|A|$ – with unequal cardinality, we can get similar results. Figure 21 shows the result when we fix the number of columns and the column cardinality, and vary the number of records only. The MPL drops quickly from the sample size 100 to 300, but keeps stable when the size increases more. This confirms our analysis that for small datasets, the entropy differences between the clusters have large variance and MPLs tend to deviate more from zero. Varying the number of columns, while fixing the other two factors, we get a similar pattern, as Figure 22 shows.

Column Cardinality We can try to understand the factor of cardinality in terms of the model complexity – column cardinality represents the inherent complexity of dataset. In general, with the increasing model complexity, the representative sample size should be increased in order to capture the complexity of the structure. On the other hand, if the sample size n keeps unchanged the increasing model complexity should bring more variance, which results in stronger noisy structures, i.e., higher MPLs. This hypothesis is supported by the experiments (Figure 23) that increasing the mean column cardinality will increase the level of MPLs with the same sample size. Figure 24 also shows that with the increasing sample size, the higher the column cardinality is, the slower the MPLs converge for both types of datasets (uniform and normal), which confirms that

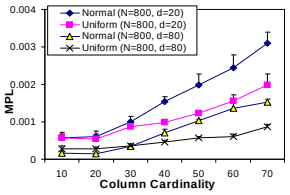

Fig. 23. Relationship between the column cardinality and MPLs

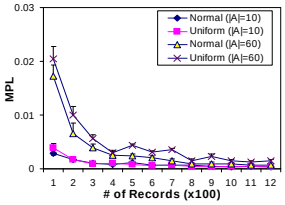

Fig. 24. The number of records vs. MPLs with two levels of column cardnality

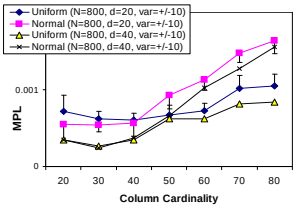

Fig. 25. unequal column cardinality vs. MPLs

we need more samples to characterize the increasing model complexity. In Figure 25, we let the column cardinalities randomly varying in the range $[|A| - 10, |A| + 10]$, but keep the mean cardinality as $|A|$. We also get a similar pattern that the MPLs increase when the mean cardinality increases.

6.6. Comparing Algorithms for Generating Approximate BKPlots

Literally, any categorical clustering algorithm that employs the same entropy minimization criterion can generate approximate BKPlots. However, the quality of approximate BKPlots can be greatly influenced by the algorithms. We compare four directly related algorithms: our proposed ACE, Monte-Carlo (Li et al., 2004), Coolcat (Barbara et al., 2002), and LIMBO (Andritsos et al., 2004) in this section, to see whether ACE is the best for generating high-quality approximate BKPlots. Monte-Carlo and Coolcat use the same criterion, "expected-entropy" that is also used by ACE, to find suboptimal partitions, while LIMBO uses mutual information in clustering, which is closely related expected-entropy. The reported results are based on ten randomly sampled datasets of each experimental data.

6.6.1. Algorithms for Generating Approximate BKPlots

Monte-Carlo Method (Li et al., 2004) 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. 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.

Coolcat (Barbara et al., 2002) 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.

We run Coolcat algorithm on each dataset with a large initial sample size (50%

of the dataset) for choosing the seed clusters and $m = 20\%$ for re-clustering, which is sufficient for improvement through re-clustering (Barbara et al., 2002). In order to reduce the effect of ordering, we also run Coolcat 20 times for each datasets and each run processes the data in a randomly generated sequence. Finally, we select the best result – having the lowest expected entropy among the 20 results.

LIMBO (Andritsos et al., 2004) algorithm is a hierarchical clustering algorithm using the Information Bottleneck (Tishby, Pereira and Bialek, 1999) criterion as the similarity between clusters. It uses a summarization structure DCF-tree to condense large datasets. In our experiments, we set the information loss factor Φ to 0, which does not use DCF-tree to compress the data. Under this setting the result is not subject to the order of records, and thus there is no randomness introduced in different runs for the same dataset.

6.6.2. Measuring Quality of BKPlots

We use three measures to evaluate the quality of approximate BKPlots.

- *Coverage Rate.* The robustness of BKPlot is represented 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 results in the BKPlot.
- *Expected Entropy.* Since the BKPlot is indirectly related to expected entropy, it might also be reasonable to check the quality of expected entropy for the partitions generated by different algorithms at the particular Ks. For a set of datasets in the same clustering structure, like DS1-i, $1 \leqslant i \leqslant 10$, we have almost same optimal clustering structure for different datasets at a fixed K. Using the mean-square-error (MSE) criterion (Lehmann and Casella, 1998) to evaluate the quality of the algorithmic result, we can decompose the errors to two parts: the deviation to the lowest expected entropy (the expected entropy of the optimal partition), and the variance of the estimated expected entropy. Let $\hat{h}$ be the expected entropy of the clustering result and h be the optimal one. $\hat{h} \geq h$ is held. 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 knowing h, if an algorithm generates clustering results with the lowest expected entropy and minimum variance among other algorithms, its BKPlots might be more trustable.

6.6.3. Results by Other Algorithms

As we have shown in Section 6.2, the BKPlots generated by ACE algorithm clearly and consistently indicate the exact Best Ks for the experimental datasets. We show some results generated by other algorithms.

The BKPlots generated by Monte-Carlo algorithm for DS1 (Figure 26) clearly identify that '3' is the best K with some small variation. However, BKPlots for DS2 show

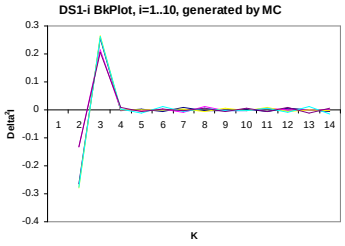

Fig. 26. BKPlots of DS1 by MC

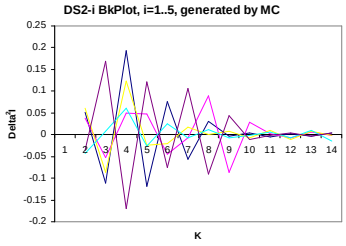

Fig. 27. BKPlots of DS2 by MC

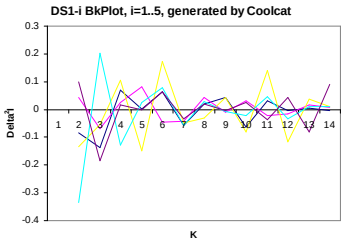

Fig. 28. BKPlots of DS1 by Coolcat

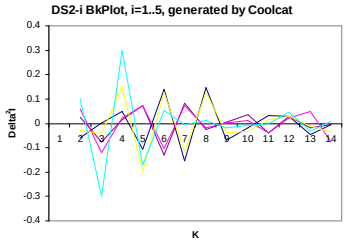

Fig. 29. BKPlots of DS2 by Coolcat

large variation on Ks. Overall, the Ks distribute from '2' to '10' for different DS2-i, not allowing the user to identify the exact Best Ks. This 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 even worse. It brings large variance for both datasets (Figure 28 and 29). The reason is that Coolcat algorithm simply does not guarantee to have near-optimal clustering results for any fixed number of clusters, and the results between different Ks do not necessarily correlate to each other.

LIMBO can successfully find the best Ks for simple structures, such as DS1 and DS2. Its DS1 and DS2 BKPlots are very similar to ACE's. However, it may generate some noise and miss some best Ks for a more complicated structure, such as Census data and DMIX data. Interestingly, it is more easily confused by higher-level structures (with less number of clusters). For example, for DMIX data (Figure 30), the LIMBO result can consistently give the best K at $K = 7$, while it generates noisy results at $K = 2, 3, 4$.

We summarize the results with the discussed measures, Coverage Rate (CR), False Discovery Rate (FDR), and expected entropy (EE) in Table 4. 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 sample datasets. In almost all cases, ACE shows the minimum expected entropy and minimum standard deviation, as well as the highest CR and lowest FDR. LIMBO is the second reliable method for generating approximate BKPlots. In general, it can generate reliable BKPlots for not-so-noisy and non-overlapping clustering structure, but may miss a part of best clustering results for overlapping clusters, such as those in DMIX and Census data. For those complicated clustering structures both MC and Coolcat will perform unsatisfactorily.

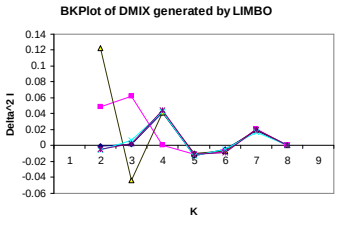

Fig. 30. BKPlots of DMIX data by LIMBO

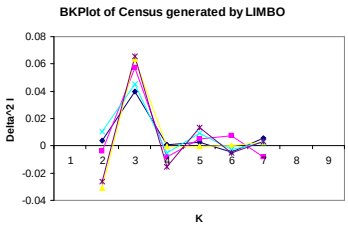

Fig. 31. BKPlots of Census data by LIMBO

		CR	FDR	EE(K=3)
DS1	ACE	100%	0%	0.283 ± 0.000
	LIMBO	100%	0%	0.283 ± 0.001
	MC	100%	0%	0.283 ± 0.001
	Coolcat	60%	85%	0.285 ± 0.005

		CR	FDR	EE($K=4$)	EE($K=6$)
DS2	ACE	100%	33%	0.218 ± 0.001	0.194 ± 0.001
	LIMBO	100%	33%	0.218 ± 0.002	0.194 ± 0.001
	MC	80%	53%	0.244 ± 0.015	0.214 ± 0.005
	Coolcat	60%	70%	0.253 ± 0.005	0.216 ± 0.008

		CR	FDR	EE($K=4$)	EE($K=7$)
DMIX	ACE	100%	0%	0.415 ± 0.003	0.293 ± 0.004
	LIMBO	90%	25%	0.415 ± 0.002	0.293 ± 0.005
	MC	20%	90%	0.494 ± 0.030	0.310 ± 0.021
	Coolcat	20%	80%	0.449 ± 0.005	0.315 ± 0.016

		CR	FDR	EE($K=2$)	EE($K=3$)
Census	ACE	100%	0%	0.398 ± 0.004	0.336 ± 0.004
	LIMBO	60%	33%	0.416 ± 0.010	0.338 ± 0.012
	MC	50%	17%	0.395 ± 0.004	0.330 ± 0.007
	Coolcat	50%	17%	0.401 ± 0.003	0.341 ± 0.007

Table 4. Quality of Approximate BKPlots

These algorithms are not likely to keep the results consistently changing through all Ks. When they cannot find near-optimal results for complicated clustering structure, it is almost impossible to generate high-quality approximate BKPlots.

7. Related Work

While many numerical clustering algorithms (Jain and Dubes, 1988; Jain and Dubes, 1999) have been published, only a handful of categorical clustering algorithms appear in literature. The general statistical analysis of categorical data was introduced in (Agresti, 1990). Although it is unnatural to define a distance function between categorical data records or to use the statistical center (the mean) of a group of categorical items, there are some algorithms, for example, K-Modes (Huang, 1997) algorithm and ROCK (Guha et al., 2000) 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, the result cannot be easily validated. Coolcat[6] has

compared the expected entropy method with ROCK and showed that expected-entropy is more appropriate for categorical data.

CACTUS (Ganti et al., 1999) adopts the linkage idea from ROCK and names it as "strong connection". The similarity is calculated by the "support" − a threshold to connect or not. A cluster is defined as a region of attributes that are pair-wise strongly connected. Still, the concept of "support" or linkage is indirect in defining the similarity of categorical data, and also makes the clustering process complicated.

Gibson et al. introduced STIRR (Gibson et al., 2000), 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 it is shown that this algorithm works for the experimental datasets having two partitions, it is challenging to determine the optimal number of clusters solely with this algorithm.

Cheng et al. (Cheng et al., 1999) applied the entropy concept in numerical subspace clustering, and Coolcat (Barbara et al., 2002) 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. Coolcat, Information Bottleneck (Tishby et al., 1999; Gondek and Hofmann, 2007), LIMBO (Andritsos et al., 2004), and the method developed in this paper are based on minimizing the same entropy criterion. The closely related work also includes Co-clustering (Dhillon et al., 2003) and Cross-association (Chakrabarti, Papadimitriou, Modha and Faloutsos, 2004). Entropy criterion is also discussed under the probabilistic clustering framework (Li et al., 2004).

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. The "Best K" problem has been discussed in terms of numerical data, especially with the mixture models. AIC and BIC (Hastie et al., 2001) are the major model-based criteria for determining the best mixture model (with the "Best K"). They have been used widely in validating the Gaussian mixture based numerical clustering. They are supposed to also be effective if appropriate mixture models are used for categorical data, among which Multinomial mixture is usually used to model categorical data. As model-based clustering fits data into assumed models, exceptions can happen when the assumed model is not appropriate. We compare the Multinomial-mixture based BIC criterion and our BKPlot method. Experimental results show that the BKPlot method has unique advantages in determining the best Ks: 1) it can determine all the best Ks for multi-layer clustering structures, while BIC cannot; 2) cluster overlapping can create certain difficulty for BIC, but BKPlot can successfully handle it.

8. Conclusion

Most of the recent research about categorical clustering has been focusing on clustering algorithms only. In this paper, we propose an entropy-based cluster validation method. Specifically, we address three problems: 1) identifying the best Ks for categorical data clustering; 2) determining whether a dataset contains significant clustering structure; 3) developing theory for handling large datasets. Our idea is to find the best Ks by observing the entropy difference between the neighboring clustering results of K and $K + 1$

clusters, respectively. The "Best-K plot (BKPlot)" is used to conveniently identify the critical Ks. BKPlots generated by different algorithm may have different performance in identify the significant clustering structures. In order to find the robust BKPlot, we also develop a hierarchical algorithm ACE based on a new inter-cluster entropy criterion "Incremental Entropy". Our experiments show that, the BKPlot method can precisely identify the Best Ks and ACE algorithm can generate the most robust BKPlots for various experimental datasets.

In addition, we also address the problems with large datasets and develop the theory of sample BKPlots for finding the best Ks for very large datasets. The result is extended to analyzing the datasets having no clustering structure. Sample no-cluster datasets are used to generate statistical tests for determining whether a given dataset has significant clustering structure.

Acknowledgements. We thank anonymous reviewers for their very useful comments and suggestions. This work was partially supported by NSF CNS, NSF CCR, NSF ITR, DoE SciDAC, DARPA, CERCS Research Grant.

Appendix

Sketch Proof of Convergence of Sample Cluster Entropy

The proof can be described in three steps.

1) estimate the distribution of any categorical value in a column of sample dataset. The cluster entropy $\hat{H}(C_i) = \sum_{j=1}^{d} \hat{H}(A_j|C_i)$ is defined by the probability p_{ijk} of each categorical value v_{jk} in column j, $1 \leq k \leq |A_j|$. Let N_i be the number of records in the cluster C_i, and N_{ijk} be the number of records containing the value v_{jk}. $p_{ijk} \sim \frac{N_{ijk}}{N_i}$. Let the random variable Y_{ijk} represent the number of records containing v_{jk} in sample cluster $C_{n,i}$. Y_{ijk} can be modeled as a *binomial distribution* $b(n_i, p_{ijk})$ (Lehmann and Casella, 1998), and thus $\frac{Y_{ijk}}{n_i}$ is an unbiased estimator for p_{ijk} with variance $\frac{p_{ijk}(1-p_{ijk})}{n_i}$, i.e., $E[\frac{Y_{ijk}}{n_i}] = p_{ijk}$ and $Var(\frac{Y_{ijk}}{n_i}) = \frac{p_{ijk}(1-p_{ijk})}{n_i}$.
2) prove that the mean of $\frac{Y_{ijk}}{n_i} \log \frac{Y_{ijk}}{n_i}$ is $p_{ijk} \log p_{ijk}$. Let $Y = \frac{Y_{ijk}}{n_i}$. Using Taylor's formula, we have $Y \log Y = Y(\log p_{ijk} + \frac{c}{\xi}(Y - p_{ijk})) = Y \log p_{ijk} + \frac{c}{\xi}(Y^2 - Y p_{ijk})$[7]. With $Y \to p_{ijk}$, $\xi \approx p_{ijk}$. Thus,

$$Y \log Y \approx Y \log p_{ijk} + \frac{c}{p_{ijk}}(Y^2 - Y p_{ijk})$$

We already know $E[Y] = p_{ijk}$ and $E[Y^2] = E^2[Y] + Var(Y) = \frac{p_{ijk}(1-p_{ijk})}{n_i} + p_{ijk}^2$. Therefore, the mean of $Y \log Y$ is given by

$$E[Y \log Y] \approx p_{ijk} \log p_{ijk} + \frac{c(1 - p_{ijk})}{n_i}$$

When $n \to N$, the item $\frac{c(1-p_{ijk})}{n_i}$ becomes very small, so $E[Y \log Y]$ converges to $p_{ijk} \log p_{ijk}$.

[7] $c = \log_d e$, d is the cardinality of the column and e is the base of the natural logarithm. ξ is a value between Y and p_{ijk}

3) prove that $E[H(C_{n,i})] \approx H(C_i)$, when $n \to N$.
Since $H(A_j|C_{n,i}) = -\sum_{k=1}^{|A_j|} \frac{Y_{ijk}}{n_i} \log \frac{Y_{ijk}}{n_i}$, we have
$E[H(A_j|C_{n,i})] \approx -\sum_{k=1}^{|A_j|} p_{ijk} \log p_{ijk} = H(A_j|C_i)$. It follows that $E[H(C_{n,i})] \approx H(C_i)$, when $n \to N$ □

Sketch Proof of Variance Estimation of Mean BKPlots

The proof can be described in four steps.

1) We will repeatedly use the following formula in the estimation. Let X_i denote the ith random variable, $i = 1 \ldots m$, $Var(X_i) \sim O(\frac{1}{n_i})$, where n_i is the sample size of X_i, a_i are some constants, and X_i and X_j are correlated with correlation coefficient ρ_{ij}, $|\rho_{ij}| \leq 1$. For simplicity, we assume that ρ_{ij} is approximately a constant, and is not related to the sample size n_i.

$$\begin{aligned} Var(\sum_{i=1}^{m} a_i X_i) &= \sum_{i=1}^{m} a_i^2 Var(X_i) + 2 \sum_{i<j\leq m} a_i a_j \rho_{ij} \sqrt{Var(X_i) Var(X_j)} \\ &\sim O(\sum_{i=1}^{m} \frac{a_i^2}{n_i} + 2 \sum_{i<j\leq m} \frac{a_i a_j}{\sqrt{n_i n_j}}) \end{aligned} \tag{6}$$

When $n_i \equiv n$, the result is simplified to $O(\frac{\sum_{i=1}^{m} a_i^2 + (\sum_{i=1}^{m} a_i)^2}{n}) \sim O(\frac{1}{n})$

2) Similar to the process of getting $E[Y \log Y]$, we apply Taylor's formula to expand $Y \log Y$, then apply the formula (6) to get $Var(Y \log Y) \sim O(\frac{1}{n} + \frac{1}{n^2}) \sim O(\frac{1}{n})$. With the Central Limit Theory (Lehmann and Casella, 1998), suppose there are s sample BKPlots, the distribution of $E[\frac{Y_{ijk}}{n_i} \log \frac{Y_{ijk}}{n_i}]$ can be approximated with a normal distribution $N(p_{ijk} \log p_{ijk}, O(\frac{1}{n_i s}))$, where $O(\frac{1}{n_i s})$ is the asymptotic notation of the variance. By the definition of column entropy and applying the formula (6), we have $Var(E[\hat{H}(A_j|C_i)]) \sim O(\frac{f(|A_j|)}{n_i s})$, where $f(|A_j|)$ is some constant determined by $|A_j|$. Let ρ'_{jk} be the correlation between $E[\hat{H}(A_j|C_i)]$ and $E[\hat{H}(A_k|C_i)]$, from the definition of cluster entropy, we have

$$\begin{aligned} Var(E[\hat{H}(C_{n,i})]) &= Var(\sum_{j=1}^{d} E[\hat{H}(A_j|C_i)]) = \sum_{j=1}^{d} Var(E[\hat{H}(A_j|C_i)]) \\ &\quad + 2 \sum_{j<k\leq d} \rho'_{jk} \sqrt{Var(E[\hat{H}(A_j|C_i)]) Var(E[\hat{H}(A_k|C_i)])} \\ &\sim O(\sum_{j=1}^{d} \frac{f(|A_j|)}{n_i s}) \sim O(\frac{1}{n_i s}) \end{aligned} \tag{7}$$

3) Since we suppose $I(n, K) \approx \frac{1}{n} IE(C_p, C_q)$ with ACE algorithm, we can estimate

the variance for $E[I(n,K)]$ as follows.

$$\begin{aligned} Var(E[I(n,K)]) &= Var\{\frac{n_p+n_q}{n}E[\hat{H}(C_p+C_q)] \\ &\quad +\frac{n_p}{n}E[\hat{H}(C_p)]+\frac{n_q}{n}E[\hat{H}(C_q)]\} \\ &\sim O(\frac{n_p+n_q}{n^2s}+\frac{n_p}{n^2s}+\frac{n_q}{n^2s}+ \\ &\quad 2(\frac{\sqrt{(n_p+n_q)n_p}}{n^2s}+\frac{\sqrt{(n_p+n_q)n_q}}{n^2s}+\frac{\sqrt{n_pn_q}}{n^2s})) \\ &\sim O(\frac{n_p+n_q}{n^2s}) \end{aligned} \tag{8}$$

To simplify it further, the variance is asymptotically $O(\frac{1}{ns})$ for $n_p+n_q \sim O(n)$.

4) By definition of $B(K)$, we have $B(n,K) = I(n,K-1) - 2I(n,K) - I(n,K+1)$, which is a linear combination of the entropy difference between partition schemes. Therefore, with formula 6, we also have

$$Var(E[B(n,K)]) \sim Var(E[I(n,K)]) \sim O(\frac{1}{ns})$$

References

Aggarwal, C. C., Magdalena, C. and Yu, P. S. (2002), 'Finding localized associations in market basket data', *IEEE Transactions on Knowledge and Data Engineering* **14**(1), 51–62.

Agresti, A. (1990), *Categorical Data Analysis*, Wiley-Interscience.

Andritsos, P., Tsaparas, P., Miller, R. J. and Sevcik, K. C. (2004), Limbo:scalable clustering of categorical data, *in* 'Proceedings of Intternational Conference on Extending Database Technology (EDBT)'.

Ankerst, M., Breunig, M. M., Kriegel, H.-P. and Sander, J. (1999), OPTICS: Ordering points to identify the clustering structure, *in* 'Proceedings of ACM SIGMOD Conference', pp. 49–60.

Barbara, D. and Jajodia, S., eds (2002), *Applications of Data Mining in Computer Security*, Klumer Academic Publishers.

Barbara, D., Li, Y. and Couto, J. (2002), Coolcat: an entropy-based algorithm for categorical clustering, *in* 'Proceedings of ACM Conference on Information and Knowledge Management (CIKM)'.

Baulieu, F. (1997), 'Two variant axiom systems for presence/absence based dissimilarity coefficients', *Journal of Classification* **14**.

Baxevanis, A. and Ouellette, F., eds (2001), *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins, 2nd edition*, Wiley-Interscience.

Bock, H. (1989), Probabilistic aspects in cluster analysis, *in* 'Conceptual and Numerical Analysis of Data', Springer-verlag.

Brand, M. (1998), An entropic estimator for structure discovery, *in* 'Proceedings Of Neural Information Processing Systems (NIPS)', pp. 723–729.

Celeux, G. and Govaert, G. (1991), 'Clustering criteria for discrete data and latent class models', *Journal of Classification* .

Chakrabarti, D., Papadimitriou, S., Modha, D. S. and Faloutsos, C. (2004), Fully automatic cross-associations, *in* 'Proceedings of ACM SIGKDD Conference'.

Chen, K. and Liu, L. (2004), 'VISTA: Validating and refining clusters via visualization', *Information Visualization* **3**(4), 257–270.

Chen, K. and Liu, L. (2005), The "best k" for entropy-based categorical clustering, *in* 'Proceedings of International Conference on Scientific and Statistical Database Management (SSDBM)', pp. 253–262.

Chen, K. and Liu, L. (2006), Detecting the change of clustering structure in categorical data streams, *in* 'SIAM Data Mining Conference'.

Cheng, C. H., Fu, A. W.-C. and Zhang, Y. (1999), Entropy-based subspace clustering for mining numerical data, *in* 'Proceedings of ACM SIGKDD Conference'.

Cover, T. and Thomas, J. (1991), *Elements of Information Theory*, Wiley.

Dhillon, I. S., Mellela, S. and Modha, D. S. (2003), Information-theoretic co-clustering, *in* 'Proceedings of ACM SIGKDD Conference'.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996), A density-based algorithm for discovering clusters in large spatial databases with noise, *in* 'Second International Conference on Knowledge Discovery and Data Mining', pp. 226–231.

Ganti, V., Gehrke, J. and Ramakrishnan, R. (1999), CACTUS-clustering categorical data using summaries, *in* 'Proceedings of ACM SIGKDD Conference'.

Gibson, D., Kleinberg, J. and Raghavan, P. (2000), Clustering categorical data: An approach based on dynamical systems, *in* 'Proceedings of Very Large Databases Conference (VLDB)', pp. 222–236.

Gondek, D. and Hofmann, T. (2007), 'Non-redundant data clustering', *Knowledge and Information Systems* **12**(1), 1–24.

Guha, S., Rastogi, R. and Shim, K. (2000), 'ROCK: A robust clustering algorithm for categorical attributes', *Information Systems* **25**(5), 345–366.

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

Hastie, T., Tibshirani, R. and Friedmann, J. (2001), *The Elements of Statistical Learning*, Springer-Verlag.

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

Jain, A. K. and Dubes, R. C. (1988), *Algorithms for Clustering Data*, Prentice hall, New York, USA.

Jain, A. K. and Dubes, R. C. (1999), 'Data clustering: A review', *ACM Computing Surveys* **31**, 264–323.

Lehmann, E. L. and Casella, G. (1998), *Theory of Point Estimation*, Springer-Verlag.

Li, T., Ma, S. and Ogihara, M. (2004), Entropy-based criterion in categorical clustering, *in* 'Proceedings of International Conference on Machine Learning (ICML)'.

Meek, C., Thiesson, B. and Heckerman, D. (2002), 'The learning-curve sampling method applied to model-based clustering.', *Journal of Machine Learning Research* **2**, 397–418.

Sharma, S. (1995), *Applied Multivariate Techniques*, Wiley&Sons, New York, USA.

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

Wang, J. and Karypis, G. (2006), 'On efficiently summarizing categorical databases', *Knowledge and Information Systems* **9**(1), 19–37.

Wrigley, N. (1985), *Categorical Data Analysis for Geographers and Environmental Scientists*, Longman.

Yu, J. X., Qian, W., Lu, H. and Zhou, A. (2006), 'Finding centric local outliers in categorical/numerical spaces', *Knowledge and Information Systems* **9**(3), 309–338.

Author Biographies

Keke Chen is currently an assistant professor in Wright State University, Dayton OH, USA. He received his PhD degree in Computer Science from the College of Computing at Georgia Tech, Atlanta GA, USA, in 2006. Keke's research focuses on distributed data intensive scalable computing, including web search, databases, data mining and visualization, and data privacy protection. From 2002 to 2006, Keke worked with Dr. Ling Liu in the Distributed Data Intensive Systems Lab at Georgia Tech, where he developed a few well-known research prototypes, such as the VISTA visual cluster rendering and validation system, the iVIBRATE framework for large-scale visual data clustering, the "Best K" cluster validation method for categorical data clustering, and the geometric data perturbation approach for service-oriented privacy-preserving data mining. From 2006 to 2008, he was a senior research scientist in Yahoo! Search&Ads Science, working on international web search relevance and data mining algorithms for large distributed datasets on cloud computing.

Ling Liu received the PhD degree in computer science in 1993 from Tilburg University, The Netherlands. She is currently an associate professor in the College of Computing at the Georgia Institute of Technology. Her research involves both experimental and theoretical study of distributed systems in general and distributed data intensive systems in particular, including distributed middleware systems, advanced Internet systems, and Internet data management. Her current research interests include performance, scalability, reliability, and security of Internet services, pervasive computing applications, as well as data management issues in mobile and wireless systems. Her research group has produced a number of software systems that are either operational online or available as open source software, including WebCQ, XWRAPElite, Omini, VISTA, and PeerCQ. She is currently on the editorial board of the International Journal of Very Large Database Systems (VLDBJ) and IEEE Transaction on Knowledge and Data Engineering. She was the program committee cochair of the 2001 International Conference on Knowledge and Information Management (CIKM 2001) held in November 2001 in Atlanta and the program committee cochair of 2002 International Conference on Ontologys, DataBases, and Applications of Semantics for Large Scale Information Systems (ODBASE), held in October, Irvine, California. She is a member of the IEEE Computer Society.

Correspondence and offprint requests to: Keke Chen, Department of Computer Science and Engineering, Wright State University, Dayton, OH 45325, USA. Email: keke.chen@wright.edu