A Contrast Pattern based Clustering Quality Index for Categorical Data

Qingbao Liu
C4ISR Technology Key Lab
National University of Defense Technology
Changsha, Hunan, China 410073
liuqingbao@nudt.edu.cn

Guozhu Dong
Department of Computer Science & Engineering
Wright State University
Dayton, Ohio, USA 45435
guozhu.dong@wright.edu

ABSTRACT
Since clustering is unsupervised and highly explorative, clustering validation (i.e. assessing the quality of clustering solutions) has been an important and long standing research problem. Existing validity measures have significant shortcomings. This paper proposes a novel Contrast Pattern based Clustering Quality index (CPCQ) for categorical data, by utilizing the *quality* and *diversity* of the contrast patterns (CPs) which contrast the clusters in clusterings. High quality CPs can characterize clusters and discriminate them against each other. Experiments show that the CPCQ index (1) can recognize that expert-determined classes are the best clusters for many datasets from the UCI repository; (2) does not give inappropriate preference to larger number of clusters; (3) does not require a user to provide a distance function.

Categories and Subject Descriptors
H.2.8 [**Database Management**]: Database Applications – *Data mining*.

General Terms
Measurement.

Keywords
Clustering validation; contrast pattern; clustering quality index.

1. INTRODUCTION
Clustering is an important data mining/analysis task, frequently used as an initial step to reveal natural groups and concepts inherent in data for many applications. Since clustering is unsupervised and highly explorative, one must rely on an objective quality measure on clusterings in order to discover the optimal clusterings hidden in the data. The qualitative evaluation of clustering solutions, or clustering validation, has been a long standing challenge in the clustering community [1~4]. Clustering validation can be based solely on the internal properties of the data (namely internal validation) or based on some external reference/validation [5]. While internal validation holds more promise for determining valid clustering results, existing internal validation indices still have serious shortcomings. Representative internal clustering quality indices in existence include distance-based, entropy-based and frequent-item-based indices, all of which have serious shortcomings. For example, when applied to many datasets from the UCI repository [6], they often fail to recognize that the expert-determined classes are the best clusters, and they often consider clusterings with larger number of clusters are better. The CPCQ index introduced here proposes a novel principle to overcome the shortcomings of existing internal quality indices. The principle asserts that a high-quality clustering should have many diversified high-quality CPs among its clusters. A *contrast pattern* (CP) is a pattern which appears with significantly varying frequencies in different clusters, thus serving to characterize their "home" (target) clusters and differentiate among clusters. The quality of individual CPs is defined in terms of their length, support, and the length of their associated closed patterns. Short CPs with long closed patterns are preferred, since short CPs indicate that the clusters are easily differentiated and identified, and long closed patterns show that the clusters are highly coherent.

Experiments indicate that the CPCQ index (1) can recognize that expert-determined classes are the best clusters for many datasets from the UCI repository; (2) does not give inappropriate preference to larger number of clusters, and (3) does not require a user to provide a distance function.

After related works, Section 2 formulates the CPCQ index. Section 3 presents an efficient method for computing CPCQ quality values. Section 4 reports experimental results. Concluding remarks and future research directions are offered in Section 5.

1.1 Related Work
There are three types of existing internal validation indices on clusterings of categorical data.

(1) Pairwise-distance-based clustering quality indices. Many pairwise-distance-based clustering quality indices have been developed; here, we review some well-known ones [7], such as Dunn's index (denoted by DCQ) [8], the Davies-Bouldin index (DBCQ) [9], the Silhouette index (SCQ) [10], and Hamming diameter/radius indices [11]. While any distance function can be used on categorical data, the Hamming distance function is often preferred.

Both the DCQ and DBCQ indices are defined as certain ratio of the minimal inter-cluster distance to the maximal intra-cluster distance. Large DCQ and small DBCQ values indicate high quality. DCQ is sensitive to noise (unstable against outliers). The SCQ index is defined using a rather complex formula, and large SCQ values are preferred. The **H**amming **Di**ameter-based **C**lustering **Q**uality index (HDiCQ) aims to minimize the cluster diameters (the maximum distance between tuples of a cluster). Smaller HDiCQ values are preferred. HDiCQ is also a noise-sensitive measure. The **H**amming **R**adius-based **C**lustering

In Proceedings of IEEE ICDM 2009.

Quality index is based on distance to cluster centers; since it is expensive (NP-complete) to compute [12], we will not consider it further. The **A**verage **H**amming **D**istance based **C**lustering **Q**uality index (AHDCQ) is defined as the weighted average of average within-cluster distances. Small AHDCQ values are preferred.

Pairwise-distance-based quality indices suffer from at least three drawbacks. (1) Since data clustering is usually performed in explorative studies, prior domain knowledge, including the "ideal" distance function, may be scarce. (2) Without an ideal distance function, users may rely on a default one that treats all attributes/items as equally important and will get poor clustering validation results. (3) Pairwise-distance-based indices often give inappropriate preference to larger number of clusters.

(2) Entropy-based clustering quality index. The **E**ntropy-based **C**lustering **Q**uality index (ECQ) emphasizes intra-cluster purity, and prefers clusterings where many of the items that occur in any given cluster are frequent items in the cluster. ECQ is equivalent to the mutual information based clustering quality index [13].

ECQ has several shortcomings. It emphasizes intra-cluster purity while ignoring inter-cluster separation. It gives strong preference to larger number K of clusters, since larger K usually leads to lower entropy. In addition, it considers only the frequency of individual items while disregarding the frequency of multi-item patterns and the different frequencies of such patterns in different clusters. Our CPCQ index addresses these shortcomings.

(3) Frequent-item-based clustering quality index. The **F**requent-**I**tem-based **C**lustering **Q**uality (FICQ) index [14] is based on the notion of frequent items. A clustering is considered good if many items of any given cluster are frequent within that cluster and there is little overlap of frequent items across clusters. Like ECQ, FICQ also considers only the frequency of individual items and disregards that of multi-item patterns. However, it may be considered as an initial step in using discrimination to measure the quality of clusterings, because it tries to minimize shared frequent items between clusters and because the frequent items within one cluster that occur infrequently in other clusters can be viewed as frequent contrast items. However, it falls short by ignoring multi-item contrast patterns, and by ignoring the diversity factor of high-quality CPs.

To avoid the drawbacks mentioned above, our CPCQ index considers a clustering as a good one if (a) there are many short CPs with high support in one cluster and low (even zero) support in the other clusters; (b) the closed CPs "equivalent" to these short CPs are long; (c) there are many highly-diverse high-quality CPs for the clusters, where high diversity is measured in terms of small item/tuple overlap among these CPs.

2. DEFINITIONS FOR CPCQ INDEX

2.1 Contrast Patterns & Equivalence Classes

We use "pattern" as synonym of "itemset." The length (the number of items) of a pattern P is $|P|$. Given a pattern P and a dataset S, $m_S(P)$ denotes the set of tuples from S that match (equivalently, contain) P, and $supp_S(P)=|m_S(P)|/|S|$ denotes the support of P in S. Given a set G of patterns, $m_S(G)$ denotes $\{t \in S \mid \exists P \in G, s.t. P \subseteq t\}$, and $supp_S(G)=|m_S(G)|/|S|$ denotes the aggregate support of patterns in G.

Let C be a cluster over a dataset D. A *contrast pattern* (also termed an "emerging pattern") [15] with C as its *target cluster* is a pattern P whose support in C is much higher than its support in D-C, i.e., $(supp_C(P)/supp_{D-C}(P)) \geq minRatio$ for some threshold *minRatio*. The ratio is defined to be ∞ if the divisor is zero.

This paper will use only CPs with infinite support ratio. This can be easily generalized to allow CPs with non-infinite support ratios. We will use a *minSupp* threshold to select those CPs whose support in their target cluster is higher than *minSupp*.

The CPCQ index will make use of an equivalence-class based structural relationship among CPs. An equivalence class [16, 17] of CPs for a cluster C in a dataset D is an ordered pair $<GS, P_{max}>$ where GS is a set of CPs and P_{max} is a CP, such that (a) each CP in GS is a subset of $P_{max,}$ and (b) no CP in GS is a superset of any other CP in GS. All patterns P in GS are equivalent in the sense that $m_D(P) = m_D(P_{max})$. Patterns in GS are the minimal-generator CPs, and P_{max} is the closed CP, of the equivalence class.

Example 1: We use a small dataset *synD* and an associated clustering $C = \{C1, C2\}$ to illustrate CPs and equivalence classes, where $C1=\{t_1, t_2, t_3\}$ and $C2=\{t_4, t_5\}$.

Table 1. synD and a clustering

tupleID	A1	A2	A3	A4	clusterID
t1	a2	b1	c2	d2	*C1*
t2	a2	b1	c1	d1	*C1*
t3	a1	b1	c1	d1	*C1*
t4	a1	b2	c2	d1	*C2*
t5	a1	b2	c1	d2	*C2*

Let *minSupp*=0.6 and *minRatio*=∞. For cluster C1, there are three equivalence classes of CPs: $EC_{11}=<GS_{11},P_{max11}>$, $EC_{12}=<GS_{12},P_{max12}>$ and $EC_{13}=<GS_{13},P_{max13}>$, where $GS_{11}=\{P_1=\{a2\}\}$, $P_{max11}=\{a2,b1\}$, $GS_{12}=\{P_2=\{b1\}\}$, $P_{max12}=\{b1\}$, $GS_{13}=\{P_3=\{c1,d1\}\}$ and $P_{max13}=\{b1, c1, d1\}$. For P_1, $m_{C1}(P_1)=\{t_1, t_2\}$ and $supp_{C1}(P_1)=2/3$. For the pattern set $G_1=\{P_1, P_3\}$, $m_{C1}(G_1)=C1$ and thus $supp_{C1}(G_1)=1$. For cluster C2, there is only one equivalence class: $EC_{21}=<GS_{21}, P_{max21}>$, where $GS_{21}=\{P_4=\{b2\}\}$, $P_{max21}=\{a1, b2\}$; moreover, $supp_{C2}(P_4)=1$.

2.2 Philosophy for CPCQ Index

Our CPCQ index is based on the rationale that a high-quality clustering should have *many diversified high-quality* CPs among its clusters. This rationale combines two key properties on CPs, namely "many diversified" and "high-quality"; these two properties help us assess the intra-cluster coherence and inter-cluster separation among clusters. We will describe how we capture "high-quality" and "many diversified" informally below, and will provide formal definitions and illustrations in Sec. 2.3.

Informally, we assess an individual CP's quality according to three factors: **(1)** The length of minimal-generator CPs. We prefer clusterings whose clusters generally have short minimal-generator CPs. If a cluster C has a very short minimal-generator contrast pattern P, then P is a strong discriminator that can be used to easily separate and distinguish C from the other clusters. The existence of short CPs implies that cluster C is significantly different from other clusters. **(2)** The length of closed contrast patterns. We prefer clusterings whose clusters generally have long closed CPs. If a cluster C has a very long closed CP P, then all the tuples in $m_C(P)$ share all items in P, implying that $m_C(P)$ is highly coherent. Technically, we will combine preferences (1) and (2) into one value, namely the ratio of the length of a closed CP over

the length of a minimal-generator CP. The minimal-generator CP and the closed CP must come from a common equivalence class of CPs. **(3)** The support of CPs. We prefer clusterings whose CPs have large support, since large support reinforces the desirability of "short discriminator" and "coherence" discussed in (1) and (2).

The abundance and high diversity of high-quality CPs is assessed through analysis of the coverage and diversity of groups of CPs. Diversity is indicated by "overlap" among multiple CPs and among multiple groups of CPs. Technically, our CPCQ index is defined in terms of the quality of a fixed number of (optimal) groups of CPs mined from the clusters of a given clustering. More specifically, abundance and diversity are evaluated according to the following three criteria: **(1)** The aggregate coverage $supp_C(G)$ by a group G of CPs of a cluster C. We prefer higher coverage, since high coverage implies the patterns in G match many (or all) of the tuples of C. **(2)** Tuple/item overlap among patterns in a group G of CPs. We prefer small overlap, since it implies that the underlying clusters have many different high-quality CPs. **(3)** Item overlap between different groups. Similarly to (2), we prefer small item overlap between different groups of CPs.

Example 2: We use the simple example in Table 2 to illustrate our philosophy. The dataset consists of eight tuples over five attributes. We can intuitively see that clustering C_1 is better than clustering C_2, since the clusters of C_1 are more individually coherent and more clearly separated than the clusters of C_2. Clustering C_1 has eight minimal-generator CPs, *{a1}, {a2}, {b1}, {b2}, {c1}, {c2}, {d1}* and *{d2},* which all are very short (with a length of 1), their closed CPs (*{a1,b1,c1,d1}* and *{a2,b2,c2,d2}*) are very long. These CPs have very high support (100%), and the minimal-generator CPs have very small pairwise item overlap. So clustering C_1 has many diversified high-quality CPs. In contrast, clustering C_2, has only two minimal-generator CPs (*{e1}* and *{e2}*), their closed CPs (*{e1}* and *{e2}*) are very short, and there are no other diversified CPs with *minSupp* >50%. So clustering C_2 has few diversified high-quality CPs. This example shows that the collection of CPs for a clustering can indeed reflect intra-cluster coherence and inter-cluster separation.

Table 2. A small dataset and two clusterings

No.	A_1	A_2	A_3	A_4	A_5	C_1	C_2
1	a1	b1	c1	d1	e1	*C1*	*C1*
2	a1	b1	c1	d1	e1	*C1*	*C1*
3	a1	b1	c1	d1	e2	*C1*	*C2*
4	a1	b1	c1	d1	e2	*C1*	*C2*
5	a2	b2	c2	d2	e2	*C2*	*C2*
6	a2	b2	c2	d2	e2	*C2*	*C2*
7	a2	b2	c2	d2	e1	*C2*	*C1*
8	a2	b2	c2	d2	e1	*C2*	*C1*

2.3 Formalizing the CPCQ Index

We first give the formula to define our quality measure on individual CPs. Suppose P is a minimal-generator CP of a given cluster C and $<GS,P_{max}>$ is *P's* equivalence class. The quality of P is defined as $QC_C(P) = supp_C(P) \times |P_{max}| / |P|$. This formula gives preference to those P which are short, have large support and long closed CPs, as discussed informally in Section 2.2.

In Example 1, the closed CP of P_1*={a2}* is P_{max11}*={a2, b1},* with $supp_{C1}(P_1)$=2/3. So $QC_{C1}(P_1) = supp_{C1}(P_1) \times |P_{max11}| / |P_1| = 2/3 \times 2/1 = 4/3$.

To formalize the "many diversified" aspect of the quality measure, we must formally define overlap between patterns, overlap among patterns in a group, and overlap between groups. Overlap can be in terms of items, or in terms of tuples.

For *item overlap*, there are three variants. The item overlap between two CPs P_1 and P_2 is defined as $ovi(P_1,P_2)=|P_1 \cap P_2|$ (the number of shared items). The item overlap of a set G of CPs is defined as $ovi(G)=avg\{ovi(P_1, P_2) \mid P_1 \in G, P_2 \in G, P_1 \neq P_2\}$ (the average item overlap between CPs of G). The item overlap between two groups G_1 and G_2 of CPs is defined as $ovi(G_1, G_2)=avg\{ovi(P_1,P_2) \mid P_1 \in G_1, P_2 \in G_2\}$.

Similarly, two variants of *tuple overlap* can be defined. The tuple overlap of two CPs P_1 and P_2 of a common target cluster C is defined as $ovt(P_1,P_2)=|m_C(P_1) \cap m_C(P_2)|$ (the number of tuples in C that match both P_1 and P_2). The tuple overlap of a set G of CPs is defined as $ovt(G)=avg\{ovt(P_1, P_2) \mid P_1 \in G, P_2 \in G, P_1 \neq P_2\}$. For a given cluster, since each group of CPs often covers the entire cluster, we do not consider the tuple overlap between groups.

We can now define the quality of a cluster C in terms of groups of CPs. Separate scenarios must be considered for a single group and for multiple groups. In the single-group case, given a group G of CPs for a target cluster C, the CP-based quality of cluster C with respect to G, denoted by $QC_G(C)$, is defined as $(\Sigma_{P \in G} QC_C(P))/((1+ovt(G)) \times (1+ovi(G)))$. In the situation involving multiple groups, given N groups $G_1,..,G_N$ of CPs for a target cluster C, the CP-based quality of cluster C in terms of the N groups, denoted by $QC_{G1..N}(C)$, is defined as $QC_{G1..N}(C) = (\sum_{i=1}^{N} QC_{G_i}(C)) / (1+avg\{ovi(G_i,G_j) \mid 1 \leq i < j \leq N\})$. We may omit the pattern groups in $QC(C)$ when it is clear from the context what $G_1,..,G_N$ are. The N groups of patterns will be selected from all CPs for cluster C in order to reflect the quality of the clusters.

We will now formalize the central concept of this paper.

Definition: Suppose $C = \{C_1, C_2, .. , C_r\}$ is a clustering of a dataset D and suppose that we have selected (through some practical and efficient method) N groups of CPs for each of the r clusters. The CP-based quality (CPCQ) index, denoted by $QC(C)$, is defined as the weighted sum of the quality of the clusters, that is, $QC(C) = \sum_{i=1}^{r} supp(C_i) \times QC(C_i)$.

3. CPCQ COMPUTATION STRATEGY

Deriving an accurate CPCQ index value depends on identification of the most representative and highest-quality groups of CPs. However, a clustering may contain a large number of CPs, leading to a large number of possible groups. It is a challenge to efficiently select the top N highest-quality groups.

3.1 The CPCQ_IncGroup Algorithm

To compute the CPCQ value of a clustering, the CPCQ ***Inc***remental ***Group***ing (CPCQ_IncGroup) algorithm proceeds in two main steps. The first step mines all the CPs and their equivalence classes from the clusters of the given clustering. This

is done by using the efficient DPMiner [18] algorithm. The second main step constructs the top N highest-quality groups of CPs and then computes the CPCQ value based on those groups.

The algorithm for the second main step constructs the top N groups incrementally for each cluster of a clustering. Suppose we have found i-1 groups. We build the ith group by adding one CP at a time, iteratively selecting a CP which, when added to the ith group, results in the greatest possible quality improvement for the groups, then adding this CP to the ith group. To give the maximum improvement, the selected CP should have large support and large closed CP/minimal-generator CP length ratio, low tuple overlap and low item overlap with all the other selected CPs, both in its own group and in the other existing groups. When the addition of CPs results in no further improvement, we stop expanding the ith group and start constructing the next group. The pseudocode of the second step of the algorithm is given in Fig. 1.

Input: C : a clustering on dataset D; ***ES***: the set of equivalence classes of CPs for some thresholds; ***N***: the number of desired top-quality groups of CPs for each cluster
Output: the CPCQ index value of the clustering C
1 FOR each cluster C of C DO
2 Let $SGP=\varnothing$; // *Set of N* Groups of contrast Patterns
3 FOR i = 1 to N DO
4 $curG=\varnothing$; $CoverofcurG=\varnothing$;
// $curG$ is the current group of contrast patterns
// $CoverofcurG$ is the set of tuples of C that match some patterns in $curG$
5 REPEAT
6 P=FindBestPatt(SGP, $curG$);
7 IF ($P\neq\varnothing$) THEN
8 $curG = curG \cup \{P\}$;
9 $CoverofcurG = CoverofcurG \cup \mathrm{m}_C(P)$;
10 END IF;
11 UNTIL ($P=\varnothing$) OR ($CoverofcurG = C$);
12 $SGP = SGP \cup \{curG\}$;
13 END FOR;
14 $QC(C)=QC_{SGP}(C)$;
15 END FOR;
16 RETURN $QC(C) = \Sigma\, supp(C) \times QC(C)$.

Figure 1. The skeleton of the CPCQ_IncGroup algorithm

The FindBestPatt() function in the CPCQ_IncGroup algorithm scans all the CPs of a given cluster C, and for each pattern P, computes the improvement to the quality of $SGP \cup \{curG\}$. The improvement is the difference in the quality of $SGP \cup \{curG\}$ before and after the addition of P to $curG$. The function finds a CP P in ES for cluster C that results in the largest quality improvement among all possible CPs for C: $QC_{SGP\cup\{curG\cup\{P\}\}}(C) - QC_{SGP\cup\{curG\}}(C) = \max\{QC_{SGP\cup\{curG\cup\{X\}\}}(C) - QC_{SGP\cup\{curG\}}(C) \mid X$ is a CP in ES for $C\}$. This function is the most time-consuming aspect of the CPCQ_IncGroup algorithm. To reduce the computational time, we use the following heuristic optimization method in Figure 2. Since all CPs in a given equivalence class have the same matching tuple set, including multiple minimal generators from one equivalence class will not give significant quality improvement. So we select at most one CP from any given equivalence class for inclusion in a given $curG$.

Output: the best contrast pattern P
1 $P=\varnothing$; $maxQCImp$=0; // initialization
2 FOR each equivalence class $<GS, P_{max}>$ of cluster C DO
3 IF $\exists X \in GS$, *s.t.* $X \in curG$ THEN CONTINUE;
4 FOR each pattern X in GS DO
5 QCImp = $QC_{SGP\cup\{curG\cup\{P\}\}}(C) - QC_{SGP\cup\{curG\}}(C)$;
6 IF ($maxQCImp$ < QCImp) THEN
7 P=X; $maxQCImp$ = QCImp;
8 END IF;
9 END FOR X;
10 END FOR $<GS,P_{max}>$;
11 RETURN P.

Figure 2. The FindBestPatt function

4. EXPERIMENTAL EVALUATION

We now use experiments to demonstrate that (1) the CPCQ index is accurate and stable; (2) the CPCQ index gives preference to neither large nor small numbers of clusters, unlike the clustering quality indexes of ECQ, AHDCQ, HDiCQ, FICQ, DCQ, DBCQ.

The accuracy and stability of CPCQ was tested using class labels provided by domain experts, the "Golden clustering," as the standard of comparison. These quality values of "Golden clustering" were compared against those clusterings obtained by other clustering algorithms. Experiments indicated that the CPCQ index often recognized the Golden clustering as superior.

Since ECQ, AHDCQ, HDiCQ, FICQ, and DBCQ prefer smaller index values, while CPCQ, DCQ, and SCQ prefer larger ones, in figures/tables below we will show the reciprocal of the CPCQ, DCQ, and SCQ index values, to facilitate comparison.

Experiments were conducted on a desktop computer with a 2.0 GHz Intel CPU and 2 GB memory running the Windows XP.

4.1 Datasets and Clustering Algorithms

Our experiments used the synthetic dataset of Example 1 and three real datasets. Three clustering algorithms (and their implementation in WEKA [19]) were used to generate clusterings.

The three datasets (from the UCI Repository) are the Mushroom (22 attributes, 8124 tuples, and two classes), Molecular Biology splice-junction gene sequences (3190 tuples, 60 attributes, and three classes), and Zoo (101 tuples and seven classes). Zoo has 15 boolean attributes and a numerical one ("legs"); we used the six values of "legs" as categorical values.

The three clustering algorithms are: (a) *EM* [20] uses K probability distributions to represent K clusters, obtained using the expectation maximization approach. WEKA's EM implementation has four parameters: *maxIteration (I)*, *numClusters* (*K*), *minStdDev* (*M*) and *seed (S)*. (b) *SimpleKMeans* [21] finds K centroids to represent K clusters. (c) *FarthestFirst* [22] is a fast variant of *SimpleKMeans*. It selects some K tuples which are fartheset from each other, instead of K random tuples, as the initial centroids. WEKA's implementation of *SimpleKMeans and FarthestFirst* uses a seed parameter (*S)*; below we list the K and S values used for them. We also note that WEKA's implementation can automatically handle categorical attributes. The three clustering algorithms were selected to generate some clusterings for comparison against the Golden clustering; they were not selected to generate optimal clusterings.

4.2 Accuracy and Stability of CPCQ Index

We now report experiments on the synthetic and two real-life datasets, to demonstrate that CPCQ is accurate, and to compare it against ECQ, AHDCQ, HDiCQ, FICQ, DCQ, DBCQ, and SCQ.

4.2.1 Experiments on a small synthetic dataset (*synD)*

We now compare clustering *C* against another clustering *C' ={C1'={t_1, t_2}, C2'={t_3, t_4, t_5 }} over synD of* Example 1. *C* is better than *C'*. Indeed, as discussed in Example 1, we know that there are four equivalence classes of CPs, for *minSupp=0.6* and *minRatio=*∞, of clustering *C*. But for clustering *C'*, there are only three equivalence classes of CPs. By placing two very similar tuples t_2 and t_3 into two different clusters, *C'* destroyed the high-quality equivalence class EC_{12} which is present in *C*. The other equivalence classes for the two clustering results are quite similar.

Table 3 shows how CPCQ (*minSupp=0.6, N=3*), ECQ, AHDCQ, HDiCQ, FICQ (*minSupp=0.6*), DBCQ, DCQ, and SCQ evaluate clusterings *C* and *C':* CPCQ, DBCQ, DCQ, and SCQ can distinguish the quality difference between the two clusterings and their judgement matches our intuition, but ECQ, AHDCQ and FICQ are unable to make this distinction, and HDiCQ renders a judgement contrary to our intuition.

Table 3. The index values on clusterings *C* and *C'*

	$CPCQ^{-1}$	ECQ	HDCQ	HDiCQ	FICQ	DCQ^{-1}	DBCQ	SCQ^{-1}
C	**0.36**	0.61	0.5	0.75	0.63	**1.5**	**2.5**	**4.24**
C'	0.46	0.61	0.5	0.5	0.63	2.0	4.0	4.35

4.2.2 Experiments on Mushroom

In this experiment, we use *FarthestFirst*, *SimpleKMean* and *EM* to create three clusterings on Mushroom, and repeated the experiments on four "*Seed*" settings (2, 4, 6 and 8). Table 4 shows how the eight indices performed on the three clusterings and the Golden clustering. It can be seen that CPCQ (*minSupp=0.01, N=5*), DCQ, and DBCQ rate the Golden clustering as the best; ECQ rates the clustering generated by *FarthestFirst-K2-S2* as the best; AHDCQ rates the clustering generated by *EM-K2* as the best; SCQ rates the clustering generated by *SimpleKMean-K2-S6* as the best; HDiCQ and FICQ (*minSupp=0.4*) rate more than one clustering as the best, and both fail to rate the Golden clustering as the best. Table 4 indicates that CPCQ is an accurate index and is better than ECQ, AHDCQ, HDiCQ, FICQ, and SCQ.

4.2.3 Experiments on Splice-junction Dataset

To further test the accuracy and stability of the CPCQ index, we conducted experiments on the Splice-junction dataset. We used WEKA's implementations of *SimpleKMean* and *EM* to create two clusterings on this dataset. *FarthestFirst* was not used, as it identified only two clusters in this dataset. Table 5 shows that the CPCQ index (*minSupp=0.04, N=5*) and the SCQ index consider the Golden clustering as the best clustering; both the ECQ and AHDCQ indices consider the clustering generated by *SimpleKMean* as the best; both the DCQ and DBCQ indices consider the clustering generated by *EM* as the best; both the HDiCQ and FICQ indices (*minSupp= 0.4*) fail to tell the difference among the three clusterings.

Table 5. Comparing index values on Splice-junction

	Golden clustering	***SimpleKMean-K3-S40***	***EM-I100-K3-S100***
$CPCQ^{-1}$	**1.277**	2.963	5.894
ECQ	0.927	**0.650**	0.927
AHDCQ	0.721	**0.507**	0.721
HDiCQ	1.0	1.0	1.0
FICQ	1.0	1.0	1.0
DCQ^{-1}	∞	30.303	**14.925**
DBCQ	∞	58.833	**25.286**
SCQ^{-1}	**32.258**	45.455	34.483

In summary, our experimental results indicate that ECQ, AHDCQ, HDiCQ, and FICQ always failed to rate the Golden clustering as the best; the assessment of the Golden clustering by DCQ, DBCQ, and SCQ varied with the dataset, and only the CPCQ index consistently recognized the Golden clustering as the highest-quality with all datasets. These experimental results demonstrate the accuracy and stability of the CPCQ index.

Table 4. Comparing the index values vs "*Seed*" settings

	Golden	**FarthestFirst: *K=2, Seed=***				**SimpleKMean:*K=2, Seed=***				**EM:*K=2,Seed=***
		2	**4**	**6**	**8**	**2**	**4**	**6**	**8**	**2~8**
$CPCQ^{-1}$	**0.021**	0.043	0.046	0.032	0.031	0.026	0.034	0.027	0.039	0.024
ECQ	0.607	**0.561**	0.595	0.637	0.652	0.569	0.594	0.573	0.625	0.602
AHDCQ	0.460	0.4456	0.462	0.513	0.514	0.456	0.451	0.450	0.495	**0.440**
HDiCQ	0.864	0.864	**0.773**	0.864	**0.773**	0.864	**0.773**	0.864	0.864	0.864
FICQ	0.98	0.99	1.00	1.00	1.00	0.99	0.98	**0.97**	1.00	**0.97**
DCQ^{-1}	**9.524**	18.868	16.949	18.868	16.949	18.868	16.949	18.868	18.868	18.868
DBCQ	**18.000**	37.000	34.000	36.000	34.000	36.000	34.000	33.000	37.000	36.000
SCQ^{-1}	5.051	3.953	3.195	6.061	4.808	4.255	3.937	**2.475**	11.236	3.610

4.3 No Preference on Number of Clusters

It is known that the ECQ index gives preference to large number (*K)* of clusters [23]. To determine whether the CPCQ index gives a similar preference, we ran *FarthestFirst* with different *K* values to produce clusterings on the Zoo dataset. The presence of seven classes in the Golden clustering in this dataset facilitates an evaluation of the impact of varying *K* on the index values for various clusterings. The CPCQ, ECQ, AHDCQ, HDiCQ, FICQ, DCQ, DBCQ, and SCQ values for these clusterings are shown in Figure 3. Clearly, this figure shows that ECQ, AHDCQ, HDiCQ, DCQ, and DBCQ give preference to larger *K*; FICQ favors smaller *K*; only CPCQ and SCQ do not give preference to larger *K*. Moreover, Figure 3 shows that CPCQ correctly determined that 7 is the best *K* for *FarthestFirst* on the Zoo dataset.

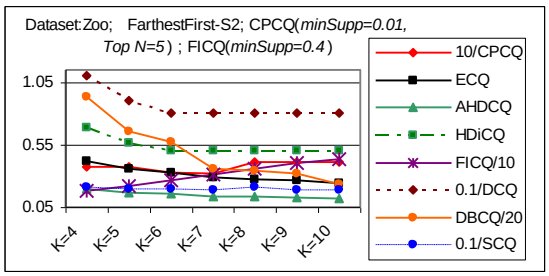

Figure 3. Index values vs. number *K* of clusters

5. CONCLUSIONS AND FUTURE WORK

This paper proposed a new internal clustering validity measure which is based on CPs. Our rationale for the CPCQ index is that a high-quality clustering should have many diversified high-quality CPs among its clusters. We have offered mathematical formulae to formalize the intuitive ideas included in this rationale. The CPCQ index overcomes the shortcomings of other popular clustering quality indices for categorical data, such as pairwise distance-based, entropy-based and frequent item-based indices, it does not give any preference to large or small numbers of clusters, and does not require the user to define a distance function or to provide domain knowledge. Experiments on synthetic dataset and real-life datasets all demonstrate that the CPCQ index is objective and scalable for clustering validation. The CPCQ index can handle high dimensional categorical datasets.

Many potential extensions of this work exist, including the extension of the CPCQ index to mixed data types and to the handling domain knowledge, and the use of this index in a novel clustering algorithm for discovering optimal clustering solutions.

ACKNOWLEDGMENTS: Work by Q. Liu was partially supported by the National Natural Science Fund of China under Grant No.70771110, and was mostly done when he was visiting Wright State University. Thanks go to the authors of [23] for the C++ code of DPMiner, and Professor Aidong Zhang for her help on improving the presentation of this paper.

REFERENCES

[1] R.C. Dubes, A.K. Jain. Validity studies in clustering methodologies. Pattern Recognition. 11(4):235–253,1979.

[2] A. K. Jain and R. C. Dubes. Algorithms for Clustering Data. Prentice-Hall, 1988.

[3] M. Halkidi, Y. Batistakis and M.Vazirgiannis. On clustering validation techniques. Intelligent Information Systems Journal, 17(2-3):107-145, 2001.

[4] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Cluster validity methods: Part I and II. SIGMOD Record, 2002.

[5] G. Brock, V. Pihur, S. Datta. clValid: An R package for cluster validation. J. of Statistical Software, 2008.

[6] A. Asuncion, D.J. Newman. UCI Machine Learning Repository. 2008.

[7] N. Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data classification. Signal Processing, 83: 825-833, 2003.

[8] J.C. Dunn. Well separated clusters and optimal fuzzy partitions. J Cybernet, 4:95–104,1974.

[9] D. Davies and D. Bouldin. A Cluster Separation Measure. IEEE Trans. Pattern Anal. Mach. Intel., 1(2), 1979.

[10] P. Rousseeuw. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. J. Comput. Appl., 20(1): 53-65, 1987.

[11] M. Bellare, O. Goldreich, M. Sudan. Free bits, PCPs, and non-approximability towards tight results. SIAM J. Comput. 27(3): 804-915, 1998.

[12] L. Gasieniec, J. Jansson, A. Lingas. Efficient approximation algorithms for the Hamming center problem. In ACM-SIAM Symposium on Discrete Algorithms. 1999.

[13] Z. He, X. Xu, S. Deng. k-ANMI: A mutual information based clustering algorithm for categorical data. Information Fusion, 9: 223–233, 2008.

[14] K. Wang., C. Yu and B. Liu. Clustering transactions using large items. CIKM, 1999.

[15] G. Dong and J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. KDD, 1999.

[16] G. Dong, C. Jiang, J. Pei, J. Li, and L. Wong. Mining Succinct Systems of Minimal Generators of Formal Concepts. DASFAA, 2005.

[17] J. Li, H. Li, L. Wong, J. Pei, and G. Dong. Minimum description length principle: Generators are preferable to closed patterns. AAAI, 2006.

[18] J. Li, G. Liu and L. Wong. Mining Statistically Important Equivalence Classes and Delta-Discriminative Emerging Patterns. KDD, 2007.

[19] www.cs.waikato.ac.nz/ml/weka/

[20] G. McLachlan and T. Krishnan. The EM algorithm and extensions. Journal of Classification, 15(1):154-156, 1997.

[21] J. MacQueen. Some methods for classification and analysis of multivariate observations. In 5th Berkeley Symposium on Mathematical Statistics and Probability. 1967.

[22] D. Hochbaum and D. Shmoys. A best possible heuristic for the k-center problem. Mathematics of OR, 10(2): 180-184, 1985.

[23] K. Chen, L. Liu. The "Best K" for entropy-based categorical data clustering. SSDBM, 2005.