Clustering Ensembles for Categorical Data

Muna Al-Razgan, Carlotta Domeniconi, and Daniel Barbará

Department of Computer Science
George Mason University
Fairfax, Virginia 22030, USA

Abstract. Cluster ensembles offer a solution to challenges inherent to clustering arising from its ill-posed nature. In this paper we focus on the design of ensembles for categorical data. Our approach leverages diverse input clusterings discovered in random subspaces. We experimentally demostrate the efficacy of our technique in combination with the categorical clustering algorithm COOLCAT.

1 Introduction

Recently, cluster ensembles have emerged as a technique for overcoming problems with clustering algorithms. It is well known that clustering methods may discover different patterns in a given set of data. This is because each clustering algorithm has its own bias resulting from the optimization of different criteria. Furthermore, there is no ground truth against which the clustering result can be validated. Thus, no cross-validation technique can be carried out to tune input parameters involved in the clustering process. As a consequence, the user is equipped with no guidelines for choosing the proper clustering method for a given dataset.

An orthogonal issue related to clustering is high dimensionality. High dimensional data pose a difficult challenge to the clustering process. Various clustering algorithms can handle data with low dimensionality, but as the dimensionality of the data increases, these algorithms tend to break down.

A cluster ensemble consists of different partitions. Such partitions can be obtained from multiple applications of any single algorithm with different initializations, or from the application of different algorithms to the same dataset. Cluster ensembles offer a solution to challenges inherent to clustering arising from its ill-posed nature: they can provide more robust and stable solutions by leveraging the consensus across multiple clustering results, while averaging out emergent spurious structures that arise due to the various biases to which each participating algorithm is tuned.

To enhance the quality of clustering results, clustering ensembles have been explored. Nevertheless, clustering ensembles for categorical data have not received much attention in the literature. In this paper we focus on the design of ensembles for categorical data. Our techniques can be used in combination with any categorical clustering approach. Here we apply our method to the partitions provided by the COOLCAT algorithm [4].

2 Related Work

We briefly describe relevant work in the literature on categorical clustering and cluster ensembles.

Clustering of categorical data has recently attracted the attention of many researchers. The k-modes algorithm [14] is an extension of k-means for categorical features. To update the modes during the clustering process, the authors used a new distance measure based on the number of mis-matches between two points. Squeezer [23] is a categorical clustering algorithm that processes one point at the time. At each step, a point is either placed in an existing cluster or it is rejected by all clusters and it creates a new one. The decision is based on a given similarity function. ROCK (Robust Clustering using links) [9] is a hierarchical clustering algorithm for categorical data. It uses the Jaccard coefficient to compute the distance between points. Two points are considered neighbors if their Jaccard similarity exceeds a certain threshold. A link between two points is computed by considering the number of common neighbors. An agglomerative approach is then used to construct the hierarchy.

To enhance the quality of clustering results, clustering ensemble approaches have been explored. A cluster ensemble technique is characterized by two components: the mechanism to generate diverse partitions, and the consensus function to combine the input partitions into a final clustering. One popular methodology utilizes a co-association matrix as a consensus function [7, 18]. Graph-based partitioning algorithms have been used with success to generate the combined clustering [22, 13, 3].

Clustering ensembles for categorical data have not received much attention in the literature. The approach in [10] generates a partition for each categorical attribute, so that points in each cluster share the same value for that attribute. The resulting clusterings are combined using the consensus functions presented in [22]. The work in [11] constructs cluster ensembles for data with mixed numerical and categorical features.

3 The COOLCAT algorithm

In this Section, we briefly present the clustering algorithm COOLCAT [4]. This algorithm has been proven effective for clustering categorical data. Thus, we use the clusterings provided by COOLCAT as components of our ensembles to further improve the grouping of data.

The COOLCAT algorithm [4] is a scalable clustering algorithm that discovers clusters with minimal entropy in categorical data. COOLCAT uses categorical, rather than numerical attributes, enabling the mining of real-world datasets offered by fields such as psychology and statistics. The algorithm is based on the idea that a cluster containing similar points has an entropy smaller than a cluster of dissimilar points. Thus, COOLCAT uses entropy to define the criterion for grouping similar objects.

Formally, the entropy measures the uncertainty associated to a random variable. Let X be a random variable with values in $S(X)$, and let $p(x)$ be the corresponding probability function of X. The entropy of X is defined as follows:

$$H(X) = -\sum_{x \in S(X)} p(x) \log(p(x))$$

The entropy of a multivariate vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_n)$ is defined as:

$$H(\boldsymbol{X}) = -\sum_{x_1 \in S(X_1)} \cdots \sum_{x_n \in S(X_n)} p(x_1, ..., x_n) \log p(x_1, ..., x_n)$$

.

To minimize the entropy associated to clusters, COOLCAT proceeds as follows. Given n points $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$, where each point is represented as a vector of d categorical values, $\boldsymbol{x}_i = (x_i^1, ..., x_i^d)$, COOLCAT partitions the points into k clusters so that the entropy of the clustering is minimized. Let $\hat{C} = \{C_1, \ldots, C_k\}$ represent the clustering. Then, the entropy associated to $\hat{C}$ is:

$$H(\hat{C}) = \sum_{j=1}^{k} \frac{|C_j|}{n} H(C_j)$$

where $H(C_j)$ is the entropy of cluster C_j:

$$H(C_j) = \sum_{x_i^1 \in S(X^1)} \cdots \sum_{x_i^d \in S(X^d)} P(x_i^1, \ldots, x_i^d | C_j) \log(P(x_i^1, \ldots, x_i^d | C_j))$$

COOLCAT uses an heuristic to incrementally build clusters based on the entropy criterion. It consists of two main phases: an initialization step and an incremental step.

During the initialization phase, COOLCAT bootstraps the algorithm by selecting a sample of points. Out of this sample, it selects the two points that have the maximum pairwise entropy, and so are most dissimilar. COOLCAT places these points in two different clusters. It then proceeds incrementally: at each step, it selects the point that maximizes the minimum pairwise entropy with the previously chosen points. At the end, the k selected points are the initial seeds of the clusters. During the incremental phase, COOLCAT constructs the k clusters. It processes the data in batches. For each data point, it computes the entropy resulting from placing the point in each cluster, and then assigns the point to the cluster that gives the minimum entropy.

The final clustering depends on the order in which points are assigned to clusters, thus there is a danger of obtaining a poor-quality result. To circumvent this problem, the authors of COOLCAT propose a reprocessing step. After clustering a batch of points, a fraction of the set is reconsidered for clustering, where the size of the fraction is an input parameter. The fraction of points that least fit the corresponding clusters is reassigned to more fitting clusters. To assess which points least match their clusters, COOLCAT counts the number of occurrences of each point's attributes in the cluster to which is assigned. This number of occurrences is then converted into a probability value by dividing it by the cluster size. The point with the lowest probability for each cluster is then removed and reprocessed according to the entropy criterion, as before. By performing this assessment at the conclusion of each incremental step, COOLCAT alleviates the risk imposed by the order of the input of points.

COOLCAT requires four input parameters: the number of clusters k, the sample size used in the initialization step, the buffer size, and the number of points considered for reprocessing.

While COOLCAT is an effective method for clustering categorical data, it still suffers from limitations inherent to clustering algorithms. The solution identified by COOLCAT depends on the initial clusters' seeding. The greedy sequential strategy used by the algorithm affects the result as well (although the reprocessing step alleviates in part this problem). Furthermore, the sparsity of the data in high dimensional spaces can severely compromise the ability of discovering meaningful clustering solutions.

In the following we address these issues by constructing ensembles of clusterings resulting from multiple runs of COOLCAT in random feature subspaces.

4 Clustering Ensemble Techniques

Consider a set $S = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ of n points. A clustering ensemble is a collection of m clustering solutions: $C = \{C_1, C_2, ..., C_m\}$. Each clustering solution C_L for $L = 1, \ldots, m$, is a partition of the set S, i.e. $C_L = \{C_L^1, C_L^2, ..., C_L^{K_l}\}$, where $\cup_K C_L^K = S$. Given a collection of clustering solutions C and the desired number of clusters k, the objective is to combine the different clustering solutions and compute a new partition of S into k disjoint clusters.

The challenge in cluster ensembles is the design of a proper consensus function that combines the component clustering solutions into an "improved" final clustering. In our ensemble techniques we reduce the problem of defining a consensus function to a graph partitioning problem. This approach has shown good results in the literature $[5, 22, 6]$.

In the next sections, we introduce our two consensus functions for categorical features using the clustering results produced by COOLCAT.

4.1 Motivation

To enhance the accuracy of the clustering given by COOLCAT, we construct an ensemble of clusterings obtained by multiple runs of the algorithm. A good accuracy-diversity trade-off must be achieved to obtain a consensus solution that is superior to the components. To improve the diversity among the ensemble components, each run of COOLCAT operates within a random subspace of the feature space, obtained by random sampling a fixed number of attributes from the set of given ones. Thus, diversity is guaranteed by providing the components different *views* (or projections) of the data. Since such views are generated randomly from a (typically) large pool of attributes, it is highly likely that each component receives a different prospective of the data, which leads to the discovery of diverse (and complementary) structures within the data.

The rationale behind our approach finds its justification in classifier ensembles, and in the theory of *Stochastic Discrimination* [16, 17]. The advantages of a random subspace method, in fact, are well known in the context of ensembles of classifiers [12, 21].

Furthermore, we observe that performing clustering in random subspaces should be advantageous when data present redundant features, and/or the discrimination power is spread over many features, which is often the case in real life scenarios. Under these conditions, in fact, redundant/noisy features are less likely to appear in random subspaces. Moreover, since discriminant information is distributed across several features, we can generate multiple meaningful (for cluster discrimination) subspaces. This is in agreement with the assumptions made by the theory of stochastic discrimination [17] for building effective ensembles of classifiers; that is, there exist multiple sets of features able to discern between training data in different classes, and unable to discern training and testing data in the same class.

4.2 Categorical Similarity Partitioning Algorithm (CSPA)

Our aim here is to generate robust and stable solutions via a consensus clustering method. We can generate contributing clusterings by running multiple times the COOLCAT algorithm within random subspaces. Thus, each ensemble component has access to a random sample of f features drawn from the original d dimensional feature space. The objective is then to find a consensus partition from the output partitions of the contributing clusterings, so that an "improved" overall clustering of the data is obtained.

In order to derive our consensus function, for each data point $\mathbf{x}_i$ and each cluster C_l, we want to define the probability associated with cluster C_l given that we have observed $\mathbf{x}_i$. Such probability value must conform to the information provided by a given component clustering of the ensemble. The consensus function will then aggregate the findings of each clustering component utilizing such probabilities.

COOLCAT partitions the data into k distinct clusters. In order to compute distances between data points and clusters, we represent clusters using *modes*. The mode of a cluster is the vector of the most frequent attribute values in the given cluster. In particular, when different values for an attribute have the same frequency of occurrence, we consider the whole data set, and choose the attribute that has the least overall frequency. Ties are broken randomly.

We then compute the distance between a point $\mathbf{x}_i$ and a cluster C_l by considering the *Jaccard distance* [19] between $\mathbf{x}_i$ and the mode $\mathbf{c}_l$ of cluster C_l, defined as follows:

$$d_{il} = 1 - \frac{|\mathbf{x}_i \cap \mathbf{c}_l|}{|\mathbf{x}_i \cup \mathbf{c}_l|} \qquad (1)$$

where $|\mathbf{x}_i \cap \mathbf{c}_l|$ represents the number of matching attribute values in the two vectors, and $|\mathbf{x}_i \cup \mathbf{c}_l|$ is the number of distinct attribute values in the two vectors.

Let $D_i = \max_l\{d_{il}\}$ be the largest distance of $\mathbf{x}_i$ from any cluster. We want to define the probability associated with cluster C_l given that we have observed $\mathbf{x}_i$. At a given point $\mathbf{x}_i$, the cluster label C_l is assumed to be a random variable from a distribution with probabilities $\{P(C_l|\mathbf{x}_i)\}_{l=1}^k$. We provide a nonparametric estimation of such probabilities based on the data and on the clustering result.

In order to embed the clustering result in our probability estimations, the smaller the distance d_{il} is, the larger the corresponding probability credited to C_l should be. Thus, we can define $P(C_l|\mathbf{x}_i)$ as follows:

$$P(C_l|\mathbf{x}_i) = \frac{D_i - d_{il} + 1}{kD_i + k - \sum_l d_{il}} \tag{2}$$

where the denominator serves as a normalization factor to guarantee $\sum_{l=1}^k P(C_l|\mathbf{x}_i) = 1$. We observe that $\forall l = 1, \ldots, k$ and $\forall i = 1, \ldots, n$ $P(C_l|\mathbf{x}_i) > 0$. In particular, the added value of 1 in (2) allows for a non-zero probability $P(C_L|\mathbf{x}_i)$ when $L = \arg\max_l\{d_{il}\}$. In this last case $P(C_l|\mathbf{x}_i)$ assumes its minimum value $P(C_L|\mathbf{x}_i) = 1/(kD_i + k - \sum_l d_{il})$. For smaller distance values d_{il}, $P(C_l|\mathbf{x}_i)$ increases proportionally to the difference $D_i - d_{il}$: the larger the deviation of d_{il} from D_i, the larger the increase. As a consequence, the corresponding cluster C_l becomes more likely, as it is reasonable to expect based on the information provided by the clustering process. Thus, equation (2) provides a nonparametric estimation of the posterior probability associated to each cluster C_l.

We can now construct the vector P_i of posterior probabilities associated with $\mathbf{x}_i$:

$$P_i = (P(C_1|\mathbf{x}_i), P(C_2|\mathbf{x}_i), \ldots, P(C_k|\mathbf{x}_i))^t \tag{3}$$

where t denotes the transpose of a vector. The transformation $\mathbf{x}_i \rightarrow P_i$ maps the d dimensional data points $\mathbf{x}_i$ onto a new space of *relative coordinates* with respect to cluster centroids, where each dimension corresponds to one cluster. This new representation embeds information from both the original input data and the clustering result.

We then define the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ as the cosine similarity between the corresponding probability vectors:

$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{P_i^t P_j}{\|P_i\|\|P_j\|} \tag{4}$$

We combine all pairwise similarities (4) into an $(n \times n)$ similarity matrix S, where $S_{ij} = s(\mathbf{x}_i, \mathbf{x}_j)$. We observe that, in general, each clustering may provide a different number of clusters, with different sizes and boundaries. The size of the similarity matrix S is independent of the clustering approach, thus providing a way to align the different clustering results onto the same space, with no need to solve a label correspondence problem.

After running the COOLCAT algorithm m times for different features we obtain the m similarity matrices $S_1, S_2, \ldots, S_m$. The combined similarity matrix

Ψ defines a *consensus function* that can guide the computation of a consensus partition:

$$\Psi = \frac{1}{m} \sum_{l=1}^{m} S_l \tag{5}$$

Ψ_{ij} reflects the average similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (through P_i and P_j) across the m contributing clusterings.

We now map the problem of finding a consensus partition to a graph partitioning problem. We construct a complete graph $G = (V, E)$, where $|V| = n$ and the vertex V_i identifies $\mathbf{x}_i$. The edge E_{ij} connecting the vertices V_i and V_j is assigned the weight value Ψ_{ij}. We run METIS [15] on the resulting graph to compute a k-way partitioning of the n vertices that minimizes the edge weight-cut[1]. This gives the consensus clustering we seek. The size of the resulting graph partitioning problem is n^2. The steps of the algorithm, which we call CSPA (Categorical Similarity Partitioning Algorithm), are summarized in the following.

Input: n points $\mathbf{x} \in \Re^d$, number of features f, and number of clusters k.

1. Produce m subspaces by random sampling f features without replacement from the d dimensional original feature space
2. Run COOLCAT m times, each time using a different sample of f features.
3. For each partition $\nu = 1, \ldots, m$:
   (a) Obtain the mode $\mathbf{c}_l$ for each cluster $(l = 1, \ldots, k)$
   (b) Compute the Jaccard distance d^ν_{il} between each data point $\mathbf{x}_i$, and each mode $\mathbf{c}_l$: $d^\nu_{il} = 1 - \frac{|\mathbf{x}_i \cap \mathbf{c}_l|}{|\mathbf{x}_i \cup \mathbf{c}_l|}$
   (c) Set $D^\nu_i = \max_l \{d^\nu_{il}\}$
   (d) Compute $P(C^\nu_l|\mathbf{x}_i) = \frac{D^\nu_i - d^\nu_{il} + 1}{kD^\nu_i + k - \sum_l d^\nu_{il}}$
   (e) Set $P^\nu_i = (P(C^\nu_1|\mathbf{x}_i), P(C^\nu_2|\mathbf{x}_i), \ldots, P(C^\nu_k|\mathbf{x}_i))^t$
   (f) Compute the similarity

   $$s^\nu(\mathbf{x}_i, \mathbf{x}_j) = \frac{P^\nu_i P^\nu_j}{\|P^\nu_i\| \|P^\nu_j\|}, \forall i, j$$

   (g) Construct the matrix S^ν where $S^\nu_{ij} = s^\nu(\mathbf{x}_i, \mathbf{x}_j)$
4. Build the *consensus function* $\Psi = \frac{1}{m} \sum_{\nu=1}^{m} S^\nu$
5. Construct the complete graph $G = (V, E)$, where $|V| = n$ and $V_i \equiv \mathbf{x}_i$. Assign Ψ_{ij} as the weight value of the edge E_{ij} connecting the vertices V_i and V_j
6. Run METIS (or spectral clustering) on the resulting graph G

Output: The resulting k-way partition of the n vertices

[1] In our experiments we also apply spectral clustering to compute a k-way partitioning of the n vertices

4.3 Categorical Bipartite Partitioning Algorithm (CBPA)

Our second approach (CBPA) maps the problem of finding a consensus partition to a bipartite graph partitioning problem. This mapping was first introduced in [6]. In [6], however, 0/1 weight values are used. Here we extend the range of weight values to [0,1].

The graph in CBPA models both instances (e.g., data points) and clusters, and the graph edges can only connect an instance vertex to a cluster vertex, forming a bipartite graph. In detail, we proceed as follows for the construction of the graph. Suppose, again, we run the COOLCAT algorithm m times for different f random features. For each instance $\mathbf{x}_i$, and for each clustering $\nu = 1, \ldots, m$, we then can compute the vector of posterior probabilities P_i^ν, as defined in equations (3) and (2). Using the P vectors, we construct the following matrix A:

$$A = \begin{pmatrix} (P_1^1)^t & (P_1^2)^t & \ldots & (P_1^m)^t \\ (P_2^1)^t & (P_2^2)^t & \ldots & (P_2^m)^t \\ \vdots & \vdots & & \vdots \\ (P_n^1)^t & (P_n^2)^t & \ldots & (P_n^m)^t \end{pmatrix} \quad (6)$$

Note that the $(P_i^\nu)^t$s are row vectors (t denotes the transpose). The dimensionality of A is therefore $n \times km$, under the assumption that each of the m clusterings produces k clusters. (We observe that the definition of A can be easily generalized to the case where each clustering may discover a different number of clusters.)

Based on A we can now define a bipartite graph to which our consensus partition problem maps. Consider the graph $G = (V, E)$ with V and E constructed as follows. $V = V^C \cup V^I$, where V^C contains km vertices, each representing a cluster of the ensemble, and V^I contains n vertices, each representing an input data point. Thus $|V| = km + n$. The edge E_{ij} connecting the vertices V_i and V_j is assigned a weight value defined as follows. If the vertices V_i and V_j represent both clusters or both instances, then $E(i, j) = 0$; otherwise, if vertex V_i represents an instance $\mathbf{x}_i$ and vertex V_j represents a cluster C_j^ν (or vice versa) then the corresponding entry of E is $A(i, k(\nu - 1) + j)$.

Note that the dimensionality of E is $(km + n) \times (km + n)$, and E can be written as follows:

$$E = \begin{pmatrix} 0 & A^t \\ A & 0 \end{pmatrix}$$

A partition of the bipartite graph G partitions the cluster vertices and the instance vertices simultaneously. The partition of the instances can then be output as the final clustering. Due to the special structure of the graph G (sparse graph), the size of the resulting bipartite graph partitioning problem is kmn. Assuming that $(km) << n$, this complexity is much smaller than the size n^2 of CSPA.

The steps of the algorithm, which we call WBPA (Weighted Bipartite Partitioning Algorithm), are summarized in the following.

Input: n points $\mathbf{x} \in \Re^d$, number of features f, and number of clusters k.

Table 1. Characteristics of the data

Dataset	k	d	n (points-per-class)
Archeological	2	8	20 (11-9)
Soybeans	4	21	47 (10-10-10-17)
Breast-cancer	2	9	478 (239-239)
Vote	2	16	435 (267-168)

```
1. Produce $m$ subspaces by random sampling $f$ features without replacement from the $d$ dimensional original feature space
2. Run COOLCAT $m$ times, each time using a different sample of $f$ features.
3. For each partition $\nu = 1, \ldots, m$:
   (a) Obtain the mode $\mathbf{c}_l$ for each cluster $(l = 1, \ldots, k)$.
   (b) Compute the Jaccard distance $d_{il}^{\nu}$ between each data point $\mathbf{x}_i$, and each mode $\mathbf{c}_l$: $d_{il}^{\nu} = 1 - \frac{|\mathbf{x}_i \cap \mathbf{c}_l|}{|\mathbf{x}_i \cup \mathbf{c}_l|}$
   (c) Set $D_i^{\nu} = \max_l \{d_{il}^{\nu}\}$
   (d) Compute $P(C_l^{\nu}|\mathbf{x}_i) = \frac{D_i^{\nu} - d_{il}^{\nu} + 1}{kD_i^{\nu} + k - \sum_l d_{il}^{\nu}}$
   (e) Set $P_i^{\nu} = (P(C_1^{\nu}|\mathbf{x}_i), P(C_2^{\nu}|\mathbf{x}_i), \ldots, P(C_k^{\nu}|\mathbf{x}_i))^t$
4. Construct the matrix $A$ as in (6)
5. Construct the bipartite graph $G = (V, E)$, where $V = V^C \cup V^I$, $|V^I| = n$ and $V_i^I \equiv \mathbf{x}_i$, $|V^C| = km$ and $V_j^C \equiv C_j$ (a cluster of the ensemble). Set $E(i, j) = 0$ if $V_i$ and $V_j$ are both clusters or both instances. Set $E(i, j) = A(i - km, j) = E(j, i)$ if $V_i$ and $V_j$ represent an instance and a cluster
6. Run METIS (or spectral clustering) on the resulting graph $G$
```

Output: The resulting k-way partition of the n vertices in V^I

5 Experimental Design and Results

In our experiments, we used four real datasets. The characteristics of all datasets are given in Table 1. The Archeological dataset is taken from [1], and was used in [4] as well. Soybeans, Breast, and Congressional Votes are from the UCI Machine Learning Repository [20].

The Soybeans dataset consists of 47 samples and 35 attributes. Since some attributes have only one value, we have removed them, and selected the remaining 21 attributes for our experiments, as it has been done in other research [8]. For the Breast-cancer data, we sub-sampled the most populated class from 444 to 239 as we have conducted in our previous work to obtain balanced data [2]. The Congressional Votes dataset contains attributes which consist of either 'yes' or 'no' responses; we treat missing values as an additional domain attribute value for each feature as conducted in [4].

Evaluating the quality of clustering is in general a difficult task. Since class labels are available for the datasets used here, we evaluate the results by computing the error rate and the normalized mutual information (NMI). The error rate is computed according to the confusion matrix. The NMI provides a measure that is impartial with respect to the number of clusters [22]. It reaches its maximum value of one only when the result completely matches the original labels. The NMI is computed according to the average mutual information between every pair of cluster and class [22]:

$$NMI = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} n_{i,j} \log \frac{n_{i,j} n}{n_i n_j}}{\sqrt{\sum_{i=1}^{k} n_i \log \frac{n_i}{n} \sum_{j=1}^{k} n_j \log \frac{n_j}{n}}} \tag{7}$$

where $n_{i,j}$ is the number of agreement between cluster i and class j, n_i is the number of data in cluster i, n_j is the number of data in class j, and n is the total number of points.

5.1 Analysis of the Results

For each dataset, we ran COOLCAT 10 times with different sets of random features. The number f of selected features was set to half the original dimensionality for each data set: $f = 4$ for Archeological, $f = 11$ for Soybeans, $f = 5$ for Breast-cancer, and $f = 8$ for Vote. The clustering results of COOLCAT are then given as input to the consensus clustering techniques being compared. (As value of k, we input both COOLCAT and the ensemble algorithms the actual number of classes in the data.)

Figures 1-8 plot the error rate (%) achieved by COOLCAT in each random subspace, and the error rates of our categorical clustering ensemble methods (CSPA-Metis, CSPA-SPEC, CBPA-Metis, and CBPA-SPEC, where SPEC is short for spectral clustering). We also plot the error rate achieved by COOLCAT over multiple runs in the entire feature space. The figures show that we were able to obtain diverse clusterings within the random subspaces. Furthermore, the instable performance of COOLCAT in the original space shows its sensitivity to the initial random seeding process, and to the order according to which data are processed. (We kept the input parameters of COOLCAT fixed in all runs: the sample size was set to 8, and the reprocessing size was set to 10.)

Detailed results for all data are provided in Tables 2-5, where we report the NMI and error rate (ER) of the ensembles, as well as the maximum, minimum, and average NMI and error rate values for the input clusterings.

In general, our ensemble techniques were able to filter out spurious structures identified by individual runs of COOLCAT, and performed quite well. Our techniques produced error rates comparable with, and sometime better than, COOLCAT's minimum error rate. CSPA-Metis provided the lowest error rate among the methods being compared on three data sets. For the Archeological and Breast-cancer data, the error rate provided by the CSPA-Metis technique is as good or better than the best individual input clustering. It is worth noticing that for these two datasets, CSPA-Metis gave an error rate which is lower than the best individual input clustering on the entire feature space (see Figures 2 and 6). In particular, on the Breast-cancer data all ensemble techniques provided

Table 2. Results on Archeological data

	Ens-NMI	Ens-ER	Max-ER	Min-ER	Avg-ER	Max-NMI	Min-NMI	Avg-NMI
CSPA-METIS	**1**	**0**	45.00	0	24.00	1	0.033	0.398
CSPA-SPEC	0.21	36.00	45.00	0	24.00	1	0.033	0.398
CBPA-METIS	0.5284	10.00	45.00	0	24.00	1	0.033	0.398
CBPA-SPEC	0.603	18.0	45.00	0	24.00	1	0.033	0.398

Table 3. Results on Soybeans data

	Ens-NMI	Ens-ER	Max-ER	Min-ER	Avg-ER	Max-NMI	Min-NMI	Avg-NMI
CSPA-METIS	0.807	**10.6**	52.1	0	24.4	1	0.453	0.689
CSPA-SPEC	0.801	12.3	52.1	0	24.4	1	0.453	0.689
CBPA-METIS	0.761	12.8	52.1	0	24.4	1	0.453	0.689
CBPA-SPEC	0.771	15.3	52.1	0	24.4	1	0.453	0.689

excellent results. For the Soybeans dataset, the error rate of CSPA-Metis is still well below the average of the input clusterings, and for Vote is very close to the average.

Also CBPA (both with Metis and SPEC) performed quite well. In general, it produced error rates comparable with the other techniques. CBPA produced error rates well below the average error rates of the input clusterings, with the exception of the Vote dataset. For the Vote data, all ensemble methods gave error rates close to the average error rate of the input clusterings. In this case, COOLCAT on the full space gave a better performance.

Overall, our categorical clustering ensemble techniques are capable of boosting the performance of COOLCAT, and achieve more robust results. Given the competitive behavior previously shown by COOLCAT, the improvement obtained by our ensemble techniques is a valuable achievement.

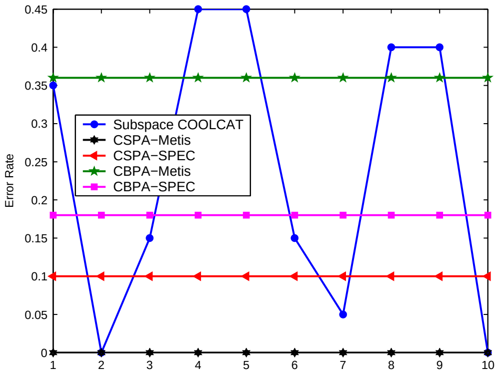

Fig. 1. Archeological data: error rates of cluster ensemble methods, and COOLCAT in random subspaces.

Table 4. Results on Breast cancer data

	Ens-NMI	Ens-ER	Max-ER	Min-ER	Avg-ER	Max-NMI	Min-NMI	Avg-NMI
CSPA-METIS	0.740	**4.3**	9.4	6.1	7.9	0.699	0.601	0.648
CSPA-SPEC	0.743	4.4	9.4	6.1	7.9	0.699	0.601	0.648
CBPA-METIS	0.723	4.8	9.4	6.1	7.9	0.699	0.601	0.648
CBPA-SPEC	0.743	4.4	9.4	6.1	7.9	0.699	0.601	0.648

Table 5. Results on Vote data

	Ens-NMI	Ens-ER	Max-ER	Min-ER	Avg-ER	Max-NMI	Min-NMI	Avg-NMI
CSPA-METIS	0.473	14.0	17.7	6.9	13.7	0.640	0.345	0.447
CSPA-SPEC	0.449	**13.5**	17.7	6.9	13.7	0.640	0.345	0.447
CBPA-METIS	0.473	14.0	17.7	6.9	13.7	0.640	0.345	0.447
CBPA-SPEC	0.439	14.2	17.7	6.9	13.7	0.640	0.345	0.447

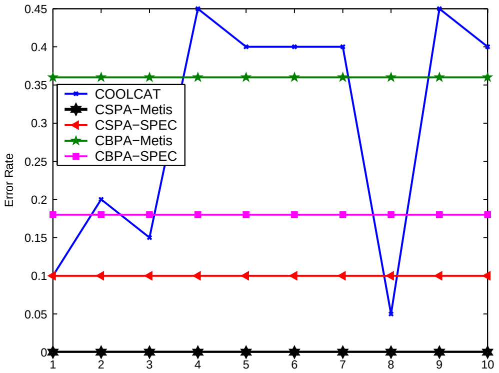

Fig. 2. Archeological data: error rates of cluster ensemble methods, and COOLCAT using all features.

6 Conclusions and Future Work

We have proposed two techniques to construct clustering ensembles for categorical data. A number of issues remains to be explored: (1) Determine which specific ensemble method is best suited for a given dataset; (2) How to achieve more accurate clustering components while maintaining high diversity (e.g., by exploiting correlations among features); (3) Test our techniques on higher dimensional categorical data. We will address these questions in our future work.

Acknowledgments This work was in part supported by NSF CAREER Award IIS-0447814.

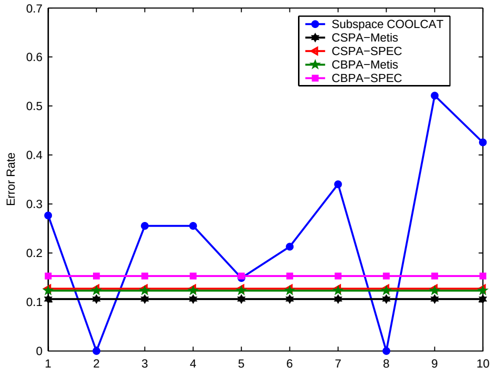

Fig. 3. Soybeans data: error rates of cluster ensemble methods, and COOLCAT in random subspaces.

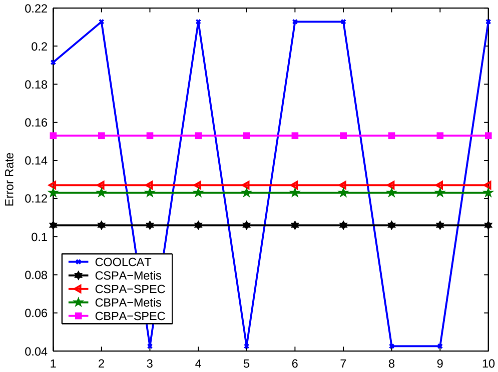

Fig. 4. Soybeans data: error rates of cluster ensemble methods, and COOLCAT using all features.

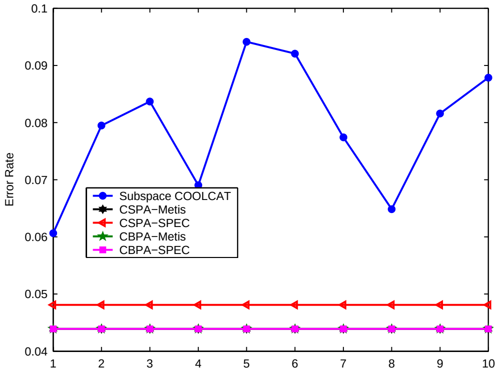

Fig. 5. Breast-cancer data: error rates of cluster ensemble methods, and COOLCAT in random subspaces.

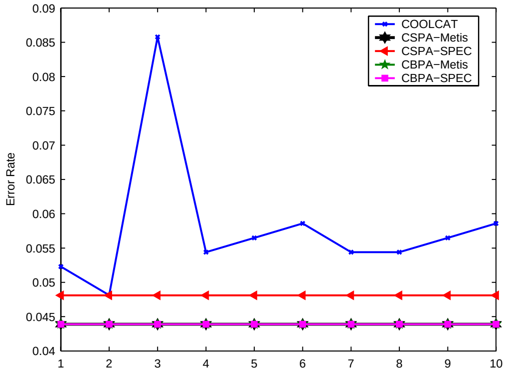

Fig. 6. Breast-cancer data: error rates of cluster ensemble methods, and COOLCAT using all features.

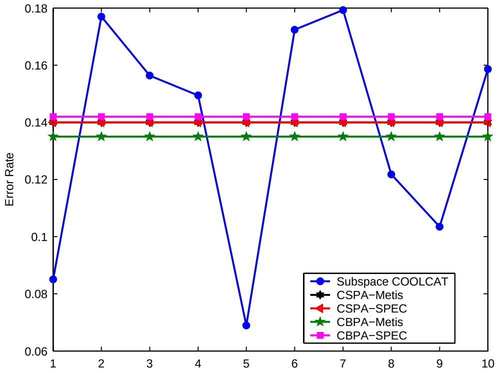

Fig. 7. Vote data: error rates of cluster ensemble methods, and COOLCAT in random subspaces.

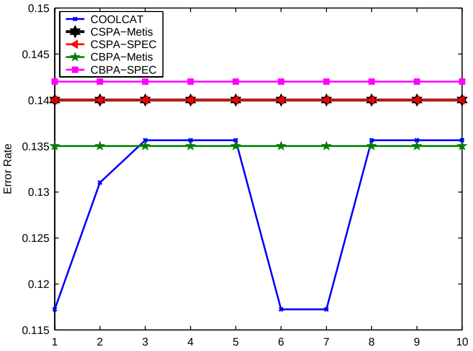

Fig. 8. Vote data: error rates of cluster ensemble methods, and COOLCAT using all features.

References

1. Aldenderfer, M. S, and Blashfield, R. K. 1984. *Cluster Analysis*, Sage Publications, No. 44.
2. Al-Razgan, M., and Domeniconi, C. 2006. Weighted clustering ensembles. In *Proceedings of SIAM International Conference on Data Mining.*
3. Ayad, H., and Kamel, M. 2003. Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors. In *Multiple Classifier Systems.* 166-175.
4. Barbará, D., Li, Y., and Couto, J. 2002. COOLCAT: an entropy-based algorithm for categorical clustering. In *Proceedings of the International Conference on Information and Knowledge Management.* ACM Press, New York, NY, 582-589.
5. Dhillon, I. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*
6. Fern, X., and Brodley, C. 2004. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the International Conference on Machine Learning.*
7. Fred, A., and Jain, A. 2002. Data clustering using evidence accumulation. In *Proceedings of the International Conference on Pattern Recognition.*
8. Gan, G., and Wu, J. 2004. Subspace clustering for high dimensional categorical data. *SIGKDD Explor. Newsl.*, 6(2), 87-94.
9. Guha, S., Rastogi, R., and Shim, K. 1999. ROCK: a robust clustering algorithm for categorical attributes In *Proceedings of Data Engineering*, 512-521.
10. He, Z., Xu, X., and Deng, S. 2005. A cluster ensemble method for clustering categorical data. *Information Fusion*, 6(2), 143-151.
11. He, Z., Xu, X., and Deng, S. 2005. Clustering Mixed Numeric and Categorical Data: A Cluster Ensemble Approach *ArXiv Computer Science e-prints*
12. Ho, T. K. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8), 832-844.
13. Hu, X. 2004. Integration of cluster ensemble and text summarization for gene expression analysis. In *Proceedings of the IEEE Symposium on Bioinformatics and Bioengineering.*
14. Huang, Z. 1998. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Min. Knowl. Discov.*, 2(3), 283-304.
15. Karypis, G., and Kumar, V. 1995. Multilevel k-way partitioning scheme for irregular graphs. Technical report, University of Minnesota Department of Computer Science and Army HPC Research Center.
16. Kleinberg, E. M. 1990. Stochastic discrimination. *Annals of Mathematics and Artificial Intelligence*, 1, 207-239.
17. Kleinberg, E. M. 1996. An overtraining-resistant stochastic modeling method for pattern recognition. *The Annals of Statistics*, 24(6), 2319-2349.
18. Kuncheva, L., and Hadjitodorov, S. 2004. Using diversity in cluster ensembles. In *Proceedings of International Conference on Systems, Man and Cybernetics.*
19. Mei, Q., Xin, D., Cheng, H., Han, J., and Zhai, C. 2006. Generating semantic annotations for frequent patterns with context analysis. In *Proceedings of the ACM SIGKDD international Conference on Knowledge Discovery and Data Mining.* ACM Press, New York, NY, 337-346.
20. Newman, D., Hettich, S., Blake, C., and Merz, C. 1998. UCI repository of machine learning databases.
21. Skurichina, M., and Duin, R. P. W. 2001. Bagging and the random subspace method for redundant feature spaces. In J. Kittler and R. Poli, editors, *Second International Workshop on Multiple Classifier Systems*, 1-10.

22. Strehl, A., and Ghosh, J. 2002. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 583–617.
23. Zengyou, H., Xiaofei, X., and Shengchun, D. 2002. Squeezer: an efficient algorithm for clustering categorical data. *J. Comput. Sci. Technol.*, 17(5), 611-624.