International Journal of Fuzzy Systems manuscript No.
(will be inserted by the editor)

Clustering of mixed data by integrating fuzzy, probabilistic and collaborative clustering framework

Arkanath Pathak · Nikhil R. Pal

Received: date / Accepted: date

Abstract Clustering of numerical data is a very well researched problem and so is clustering of categorical data. However, when it comes to clustering of data with mixed attributes, the literature is not that rich. For numerical data, fuzzy clustering, in particular, the fuzzy c-means (FCM), is a very effective and popular algorithm, while for categorical data, use of mixture model is quite popular. In this paper, we propose a novel framework for clustering of mixed data which contains both numerical and categorical attributes. Our objective is to find the cluster substructures that are *common* to both the categorical and numerical data. Our formulation is inspired by the FCM algorithm (for dealing with numerical data), mixture models (for dealing with categorical data) and the collaborative clustering framework for aggregation of the two - it is an integrated approach that judiciously uses all three components. We use our algorithm on a few commonly used datasets and compare our results with those by some state of the art methods.

Keywords Fuzzy Clustering · Mixed Data · Mixture Models · Collaborative Clustering

1 Introduction

Clustering is one of the most commonly used exploratory data analysis techniques. In majority of the real life data sets, objects are usually represented by numerical features (we call such data sets as objects data sets). There are also many examples, where objects are represented by categorical values (we call such data sets as categorical data sets). For both object data and categorical data, there are many clustering algorithms available [19,12,18,17,24,16]. Of course, the number of algorithms available for object data are much more than that for the categorical data. In addition to these two types of data, often we get mixed-data where an object is represented by both numerical features as well as categorical attributes. Although, there are several clustering algorithms for mixed data, the literature is not as rich as it is for the other two types of data. In this study we shall focus on clustering of mixed data. The available spectrum of clustering algorithms can be divided into hard and soft (fuzzy or probabilistic) clustering algorithms. In case of a hard clustering algorithm, such as the k-means [11], an object either belongs to a cluster or does not belong; while for a fuzzy or probabilistic cluster, an object can belong to more than one cluster and its degree of belonging to a cluster is represented either by a probability or membership value [4, 3]. In this paper we shall develop a fuzzy (soft) clustering framework to find a partition of a mixed data set by exploiting the common cluster substructure present in both categorical and numeric data.

Let $X = \{x_1, x_2, ..., x_n\}$ be a dataset which is to be clustered into c clusters. There are many clustering algorithms in the literature. Fuzzy c-means (FCM) [4]remains one of the most widely used algorithms for clustering datasets that deal with numerical data. The algorithm tries to minimize the following objective function:

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^m d_{ij} \quad , m > 1 \tag{1}$$

Arkanath Pathak
Indian Institute of Technology Kharagpur, Kharagpur, West Bengal, India 721302
Tel.: +91-8967024282
E-mail: pathak.arkanath@gmail.com

Nikhil R. Pal
Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta, West Bengal, India 700108
Tel.: +91-33-25752906
E-mail: nikhil@isical.ac.in

subject to the following set of constraints:

$$\sum_{i=1}^{c} u_{ij} = 1, \quad \forall j; 1 < \sum_{j=1}^{n} u_{ij} < n, \quad \forall i; u_{ij} \in [0,1] \tag{2}$$

where n is the number of data points and c is the number of clusters. u_{ij} denotes the belongingness of the j^{th} data point to the i^{th} cluster, and $d_{ij} = D_{ij}^2$ where D_{ij} is distance between the center of the i^{th} cluster and the j^{th} data point. Usually D_{ij} is an inner product induced distance including the Euclidean distance. In this investigation we shall use $m = 2$. The details of the FCM algorithm can be found in [2].

Although in many cases we need to deal with only numerical data, it is not uncommon to have others data types; the other most commonly encountered data type is the categorical data type. In this paper, we devise a new FCM type algorithm for clustering of data that contain both numerical and categorical attributes. In the past researchers have proposed various approaches for clustering mixed datasets [1,6,9,13,20,21,26,27,22]. Ahmad and Dey[1] proposed a k-means like algorithm for clustering of mixed data which uses a distance measure involving two parts: one for the numerical attributes and the other for the categorical attributes. To make the distance between categorical attributes useful, the distance between two different attribute values is computed based on their overall distribution as well as their co-occurrence with other attributes. However, this algorithm does not allow fuzzy membership of the data points. Huang[15] proposed a k-prototype algorithm for mixed datasets that combines the k-means algorithm and k-modes algorithm. Here to compute the distance between the categorical prototype and the data vector (categorical part) the number of mismatch in attribute values is considered. For each cluster, a weight for the categorical attributes is also used. Ji et al.[20] extended the a k-prototype algorithm to include fuzzy partitions. In addition, authors use a concept of fuzzy centroid to represent the prototype of a cluster, where a in a fuzzy centroid each attribute has a fuzzy category value.

One of the most efficient partitioning strategies is the use of mixture models. A mixture model assumes each cluster to be generated from some specific probability distribution. The objective is then to find the mixture parameters by maximizing the likelihood of a given data set. The log likelihood in such an algorithm is of the following form ([5]):

$$L = \sum_{j=1}^{n} \log \sum_{i=1}^{c} (\pi_i p(\mathbf{x_j}|\theta_\mathbf{i})) \tag{3}$$

where c is the number of clusters, n is the number of data members, π_i are the cluster proportions and θ is the vector representing the distribution parameters. If we assume that X is incomplete and assume the unobserved data as $Y = \{y_i\}_{i=1}^{n}$; $y_i \in \{1, 2, \cdots, c\}$; where $y_i = k$ if x_i is from the k^{th} cluster, then the log of the complete likelihood is:

$$\log P(X, Y|\theta) = \sum_{i=1}^{n} \log(P(x_i|y_i)P(y_i)) = \sum_{i=1}^{n} \log(\pi_{y_i} P(x_i|\theta_{y_i})). \tag{4}$$

The EM algorithm is used to maximize the likelihood with respect to it's parameters. Numerical attributes are most commonly modelled as mixtures of Gaussian distributions.

Everitt[9] proposed such an algorithm by relating both numerical and ordinal attributes to Gaussian distributions. Here the ordinal values are assumed to be generated by thresholding from some unobservable continuous variables. However, this assumption often is not realistic. Jorgensen and Hunt[21] proposed another mixture model algorithm, which generalizes both the latent class model and the multivariate normal mixture model. This mixture model can also be viewed as a graphical model. Gath and Geva[10] proposed an FCM type algorithm by associating the dissimilarity measure d_{ij} in the fuzzy c-means algorithm to an "exponential" distance measure. This method makes an implicit assumption that the clusters can be modeled using Gaussian distributions. The "exponential" distance measure used is defined as the negative log-likelihood of the Gaussian distribution representing the cluster:

$$d_{ij} = -\log p(\mathbf{x_j}|\theta_\mathbf{i}) \tag{5}$$

Chatzis[6] recently extended upon the approach used by Gath,Geva[10] to include the categorical attributes. Chatzis modelled the categorical attributes of the observations for each cluster as a multinomial distribution. Assuming the categorical part to be independent of the numerical part, the joint probability takes the following form:

$$p(\mathbf{x_j}|\theta_\mathbf{i}) = \mathcal{N}(\mathbf{x_j^N}|\mu_\mathbf{i}, \mathbf{\Sigma_i})\mathbf{Mult}(\mathbf{x_j^C}|\gamma_\mathbf{i}) \tag{6}$$

where $\mathbf{x_j}$ is the concatenation of numerical feature vector $\mathbf{x_j^N}$ and the categorical feature vector $\mathbf{x_j^C}$, i.e. $\mathbf{x_j} = [\mathbf{x_j^N} : \mathbf{x_j^C}]$; $\mathcal{N}(\mathbf{x_j^N}|\mu_\mathbf{i}, \mathbf{\Sigma_i})$ is a Gaussian distribution modeling the numerical part of the data for cluster i and $\mathbf{Mult}(\mathbf{x_j^C}|\gamma_\mathbf{i})$ represents a multinomial distribution for the categorical part of the data in the i-th cluster. The objective function proposed by Chatzis is as follows:

$$J_\lambda = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} d_{ij} + \lambda \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} \log \frac{u_{ij}}{\pi_i}, \tag{7}$$

where $d_{ij} = p(\mathbf{x_j}|\theta_\mathbf{i})$ and u_{ij} is the membership value of the j-th data point to the i-th cluster. The regularization term in the objective function is inspired from the work of Honda and Ichihashi[14], which is related to the fuzzy entropy of the partition matrix, and is used as a substitute for the fuzzifier that is normally present in FCM-type algorithms. The algorithm estimates the parameters of the Gaussian and multinomial distributions using an EM-type algorithm iterating through the first order necessary conditions of optima of the objective function derived using the Lagrange multiplier techniques.

2 Proposed Approach

We propose an entirely different approach which uses the FCM type objective function for clustering of mixed data. Our objective is to partition the data exploiting the cluster substructures that are common to both the categorical and numeric part of the data.

Our approach is inspired by the collaborative fuzzy clustering framework introduced by Pedrycz [23], but we solve here a different problem. Collaborative clustering deals with finding a common cluster structure among multiple data sets. In a collaborative clustering framework, typically data are collected in a distributed manner at different sites. In its simplest form, the objective of the data analyst is to find cluster structures that are common to all data sets. But because of security, privacy or other reasons the data sets cannot be shared between sites. However, the clustering results found independently at different data sites can be shared. There is a good amount of literature on collaborative fuzzy clustering [25, 23, 8]. To illustrate the idea, consider two sites A and B associated with data sets X_A and X_B. In [25], to find clusters in X_A, a regularizing term is added to the usual FCM objective function defined on X_A, where the regularizing term uses a weighted sum of the squared difference between the membership value produced by X_A and that computed using the centroids generated by X_B at site B. As mentioned earlier we solve a different problem where the original data set separated into two data sets, one involving only numerical features and the other involving only categorical features. We also use a regularizing term involving the difference between the two partition matrices produced on the two data sets, but we do not use the distance of a data point from a centroid as a weight as done in [25].

2.1 Partitioning data

We partition the dataset into two separate parts, one corresponding to the numerical attributes and the other corresponding to the categorical attributes, denoted as $\mathbf{x^N}$ and $\mathbf{x^C}$ respectively. We shall cluster the numerical dataset separately and then cluster the categorical dataset using the partitions obtained from the numerical dataset as a constraint. This process will then be reversed. We, thus, also have two different partition matrices, U^N and U^C. For obvious reasons, the number of clusters in both datasets should be the same. What we now intend to achieve is the following: these two partitioning to be as good as possible and at the same time as similar as possible. In other words we want to find the cluster structure that is common to both numerical and categorical data. Hence, we need to include this factor in the objective functions of the two datasets, X^N and X^C. We add a regularizing term to each of the cost functions that captures the dissimilarity between the two partition matrices. The cost functions for the two datasets take the following form:

$$J_\lambda^n = \sum_{i=1}^{c}\sum_{j=1}^{n} {u_{ij}^N}^2 d_{ij}^N + \lambda \sum_{i=1}^{c}\sum_{j=1}^{n} (u_{ij}^N - u_{ij}^C)^2 \tag{8}$$

$$J_\lambda^c = \sum_{i=1}^{c}\sum_{j=1}^{n} {u_{ij}^C}^2 d_{ij}^C + \lambda \sum_{i=1}^{c}\sum_{j=1}^{n} (u_{ij}^N - u_{ij}^C)^2 \tag{9}$$

where:

$$d_{ij}^N = ||\mathbf{x_j^N} - \mathbf{v_i}||^\mathbf{2} \tag{10}$$

$$d_{ij}^C = -\log p(\mathbf{x_j^C}|\gamma_\mathbf{i}) \tag{11}$$

In (10), $\mathbf{v_i}$ is the center of the i^{th} cluster, and $||\mathbf{x_j^N} - \mathbf{v_i}||$ is the euclidean distance between the numerical part of the j^{th} data point and $\mathbf{v_i}$. The probabilistic distance measure in (11) is the same as that used by Chatzis[6], which is described below:

$$p(\mathbf{x_j^C}|\gamma_\mathbf{i}) = \mathbf{Mult}(\mathbf{x_j^C}|\gamma_\mathbf{i}) = \prod_{\mathbf{k=1}}^{\mathbf{m_c}} \prod_{\mathbf{l=1}}^{\mathbf{L_k}} {\gamma_{\mathbf{ikl}}}^{\mathbf{1}-\delta(\mathbf{x_{jk}^C},\mathbf{l})} \tag{12}$$

where m_c is the number of categorical attributes, L_k is the number of possible values for the k^{th} categorical attribute. Here $\gamma_{\mathbf{ikl}}$ is the probability of the k^{th} categorical attribute taking the l^{th} possible value for the i^{th} cluster's distribution. $\delta(x_{jk}^C, l) = 0$ if the k^{th} categorical attribute of the j^{th} data member takes the l^{th} possible value, otherwise $\delta(x_{jk}^C, l) = 1$.

2.2 The algorithm and the update equations

Using the Lagrange Multiplier method, we can derive the expressions for the update of the parameters. The constraints on each of the two partitions are the following:

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j \tag{13}$$

and those for the categorical parameters are:

$$\sum_{l=1}^{L_k} \gamma_{i,k,l} = 1, \forall i, k \tag{14}$$

Using these constraints, we arrive at the following set of update equations for the numerical part:

$$u_{ik}^N = \frac{1}{\sum_{j=1}^{c} \frac{d_{ik}^N + \lambda}{d_{jk}^N + \lambda}} + \lambda \frac{\sum_{j=1}^{c} \frac{u_{ik}^C - u_{jk}^C}{d_{jk}^N + \lambda}}{\sum_{j=1}^{c} \frac{d_{ik}^N + \lambda}{d_{jk}^C + \lambda}} \tag{15}$$

$$\mathbf{v_i} = \frac{\sum_{j=1}^{n} {u_{ij}^N}^2 \mathbf{x_j^N}}{\sum_{j=1}^{n} {u_{ij}^N}^2} \tag{16}$$

and the following for the categorical part:

$$u_{ik}^C = \frac{1}{\sum_{j=1}^{c} \frac{d_{ik}^C + \lambda}{d_{jk}^C + \lambda}} + \lambda \frac{\sum_{j=1}^{c} \frac{u_{ik}^N - u_{jk}^N}{d_{jk}^C + \lambda}}{\sum_{j=1}^{c} \frac{d_{ik}^C + \lambda}{d_{jk}^C + \lambda}} \tag{17}$$

$$\gamma_{ikl} = \frac{\sum_{j=1}^{n} {u_{ij}^C}^2 (1 - \delta(x_{jk}^C, l))}{\sum_{h=1}^{L_k} \sum_{j=1}^{n} {u_{ij}^C}^2 (1 - \delta(x_{jk}^C, h))} \tag{18}$$

Using the above update equations (15) and (16), we can minimize the cost function at (8) when we are given a partition matrix computed based on the categorical attributes. Similarly, using (17) and (18), we can optimize (9).

We first propose an algorithm (**Algorithm 1**) for clustering the numerical data, which also maximizes the closeness between the partition generated by the numerical data and a given partition generated by the categorical data. Following that, we provide another algorithm (**Algorithm 2**) that does the reverse job of clustering the categorical data. However, our goal is to find a common structure between the two partitions minimizing $J_\lambda^n + J_\lambda^c$. Thus another plausible choice of objective function to drive the clustering algorithm would be as follows:

Minimize

$$J_\lambda^{nc} = \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}^N}^2 d_{ij}^N + \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}^C}^2 d_{ij}^C + \lambda \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{ij}^N - u_{ij}^C)^2 \tag{19}$$

subject to (13) and (18). It is easy to verify that the necessary conditions for minimizations are nothing but equations (15), (16), (17) and (18). Based on the optimization of J_λ^{nc}, we propose an approach (**Algorithm 3**). Note that the algorithm will converge if any of the U^C, U^N, V or γ converges.

Algorithm 1 Generates a partition based on the numerical attributes that matches the clustering produced by the categorical attributes

1: **if** parameter U^C for the Categorical Partition is not provided **then**
2: Randomly initialize U^C and γ to satisfy (13),(14)
3: Calculate J_λ^C using (9) with $\lambda = 0$, store it as J_{old}^C
4: Update u_{ij}^C & γ using (17) & (18) with $\lambda = 0$
5: Calculate J_λ^C using (9) with $\lambda = 0$, store it as J_{new}^C
6: **if** $|J_{new}^C - J_{old}^C| < \epsilon$ **then**
7: goto Step 11
8: **else**
9: $J_{old}^C \leftarrow J_{new}^C$
10: goto Step 4
11: **if** parameters U^N or $\mathbf{V^N} = (\mathbf{v_1^N}, ...\mathbf{v_c^N})$ for the Numerical data is not provided **then**
12: Randomly initialize $\mathbf{V^N}$ or U^N to satisfy (13)
13: Calculate J_λ^N using (8), with the provided λ, store it as J_{old}^N
14: Update the u_{ij}^N and $\mathbf{V^N}$ values using (15) and (16)
15: Calculate J_λ^N using (8), store it as J_{new}^N
16: **if** $|J_{new}^N - J_{old}^N| < \epsilon$ **then**
17: **return** U^N and $\mathbf{V^N}$
18: **else**
19: $J_{old}^N \leftarrow J_{new}^N$
20: goto Step 14

2.3 Choice of λ

The parameter λ in Algorithms 1 and 2 controls the extent of impact of the other partition on the clustering of data. For instance, choosing λ as 0 makes Algorithm 1 work exactly like the FCM, and choosing λ as something very large like 10000 will give rise to a numerical partition structure which is highly similar to the existing categorical partition. A similar effect will be present for Algorithm 2. Therefore, we can choose λ in these two algorithms based on our need. In Algorithm 3, however, chosing λ is not an easy task. A larger value of λ implies a greater similarity between U^C and U^N, and thus, the algorithm may converge faster. However, very large values of λ may discount the importance of individual objective functions. In our experimental evaluations, $\lambda = 100$ gave useful results. An alternative approach for choosing λ in each of the three algorithms can be to use a small value at the beginning of the iterations and then increase it up to some value with iterations. We shall refer to this approach as the *dynamic λ approach*.

3 Experimental observations

3.1 Synthetic Data

To evaluate the performance of the algorithms, and to give a basic understanding of what the algorithms

Algorithm 2 Generates a partition based on the categorical attributes that matches the clustering produced by the numerical attributes

if parameter U^N for the Numerical Partition is not provided **then**
  Randomly initialize U^N to satisfy (13) & compute $\mathbf{V^N} = (\mathbf{v_1^N}, ...\mathbf{v_c^N})$
  Calculate J_λ^N using (8) with $\lambda = 0$, store it as J_{old}^N
  Update u_{ij}^N & $\mathbf{V^N}$ using (15) & (16) with $\lambda = 0$
  Calculate J_λ^N using (8) with $\lambda = 0$, store it as J_{new}^N
  if $|J_{new}^N - J_{old}^N| < \epsilon$ **then**
    goto Step 11
  else
    $J_{old}^N \leftarrow J_{new}^N$
    goto Step 4
if parameters U^C or γ for the Categorical Partition are not provided **then**
  Randomly initialize the U^C or γ matrices to satisfy (13),(14)
Calculate J_λ^C using (9), with the provided λ, store it as J_{old}^C
Update the u_{ij}^C and γ values using (17) and (18)
Calculate J_λ^C using (9), store it as J_{new}^C
if $|J_{new}^C - J_{old}^C| < \epsilon$ **then**
  return U^C and γ
else
  $J_{old}^C \leftarrow J_{new}^C$
  goto Step 14

do, we have implemented the algorithms on a synthetic dataset along with some of its noisy variants. The synthetic dataset contains 400 points divided into two clusters. We call this dataset *Synth*. This dataset contains two numerical features and four categorical attributes. The numerical part is generated by a mixture of two 2-D Gaussian distributions with mean vectors $[1, 2]$ and $[4, 5]$. The covariance matrix for each cluster is $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Each cluster contains 200 data points. The scatter plot of the numerical part of the dataset is shown in Fig. 1.

The four categorical attributes A_1, A_2, A_3, A_4 can take 3, 2, 4 and 3 possible values respectively. To test our algorithms, we shall add various levels of noise on the categorical part of the data.

3.1.1 No noise

In the case of no noise, we have generated the categorical data having the same cluster structure as that of the numerical partition; in order to achieve this we assign a fixed vector of attribute values for the first cluster, while for the points from the other cluster we associate a fixed but different vector of attribute values. These two vectors were chosen as $[2, 1, 2, 1]$ and $[1, 1, 4, 2]$. For plotting the categorical attributes we have used orthogonal coding. Whenever we need to plot more than two

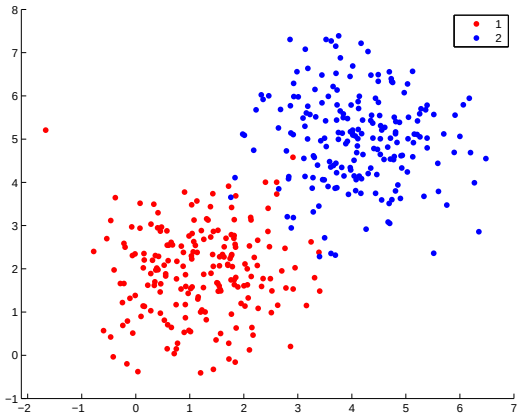


Fig. 1: The numerical feature part of the Synthetic data

attributes, we performed PCA and chose the top two principle components to plot.

Algorithm 3 Mixed Clustering for both Partitions

Run **Algorithm 1** and obtain $U^N(0)$ and $\mathbf{V^N}(0)$ matrices
Run **Algorithm 2** and obtain $U^C(0)$ and $\gamma(0)$ matrices
$t = 1$
Update the $U^N(t)$, $\mathbf{V^N}(t)$, $U^C(t)$ and $\gamma(t)$ values using (16),(17),(18) and (19)
Calculate $tolerance = ||U^C(t) - U^C(t-1)||$
if $tolerance < \epsilon$ **then**
  return $(U^C(t) + U^N(t)/2)$
else
  $t = t + 1$
  goto Step 4

Algorithm 1 gave no mislabels for both $\lambda = 100$ and the dynamic λ approach. Algorithm 2 gave around 9 mislabels. This behaviour explains that the numerical part exhibits a cluster structure with some overlap that has resulted about 9 mislabels on average. Algorithm 3 performed a perfect clustering yielding no mislabels for all parameter settings that we have tried.

3.1.2 Results with different levels of noise

We have generated five noise corrupted versions of dataset Synth by adding different levels of noise only to the categorical attribute values and observed their effect on the performance of our algorithms. For adding noise, we randomly chose some percentage of the categorical attribute values (in this case there are a total of 1600 attribute values), and randomly changed them to any of the possible values (including the existing value) for the respective attribute. We considered five different

levels of noise : 10%, 20%, 30%, 40% and 50% and these five datasets are named as Synth10, Synth20, Synth30, Synth40 and Synth50 respectively.

As illustrations, in Fig. 2, in the left hand side panels, we display scatter-plot of the top two principal components of only the categorical part of the data and in the right hand side, the top two principal components of the entire dataset. The two clusters (the actual clusters, not the output of our algorithms) are represented by two different colors as well as by two different styles for 10% and 50% noise levels. From these figures, it is clear that categorical data get mixed up more with increase in noise level as we corrupt the categorical part.

For Synth10, Algorithm 1 gave one mislabelling for both $\lambda = 100$ and the dynamic λ approach. For the strategy with dynamic λ, in this case as well as in other cases as applicable, we increase λ as 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50 with iterations and then fix it at 100. Algorithm 2 gave 10.16 mislabels for $\lambda = 100$ and 11.64 mislabels with the dynamic λ approach. This result explains that Algorithm 1 gets affected by adding noise in the categorical part. Algorithm 3 performed perfect again by giving no mislabels for both parameter settings. Table 1 summarizes the clustering results with Algorithms 1 and 2 for different noise levels. For each noise level we ran the algorithm 30 times and Table 1 depicts the average number of points that are wrongly clustered over those 30 runs. As the noise level increases, there is a marginal decrease in the performance of both algorithms for both choices of λ, fixed and dynamic. Algorithm 3, gives no mislabels up to 20% noise, and results in much lesser mislabels compared to the Algorithms 1 and 2 for the cases with more noise (See Table 2)

Table 1 reveals that with increase in the noise level, in general the misclassification increases. For Algorithm 1, on an average, the mislabeling increases initially by about 0.25% and then approximately 3% with every additional 10% noise and this is true for both fixed λ and dynamic λ. For Algorithm 2, on the other hand, from no-noise case up to 30% noise there is not much impact of noise on the performance of the algorithm and beyond 20% noise there is a marginal increase in misclassification. In case of Algorithm 2, generally the dynamic λ is found to work better.

Table 2 summarizes the results produced by Algorithm 3 on different variants of Synth for $\lambda = 100$ and dynamic λ approach summarized over 30 runs. For Algorithm 3, both fixed and dynamic λ are found to work equally well. Even when the noise level is 50%, only 1.5-1.7% mislabeling is produced by Algorithm 3. Note that, when the algorithm converges, both U^N and U^C converges to the same partition and hence both criteria become the same.

Table 1: Percentage (%) of mislabeling for synthetic data for Algorithms 1 and 2

Noise	Algorithm 1		Algorithm 2	
(%)	$\lambda = 100$	λ^*	$\lambda = 100$	λ^*
0	0.00	0.00	2.29	2.48
10	0.25	0.25	2.54	2.91
20	0.50	0.50	2.76	2.61
30	3.00	3.00	3.22	3.20
40	5.00	5.00	4.53	3.14
50	8.25	8.25	4.57	4.32

Table 2: Percentage (%) of mislabeling for synthetic data with Algorithm 3

Noise	$\lambda = 100$			λ^*		
(%)	U^N	U^C	$U^C + U^N$	U^N	U^C	$U^C + U^N$
0	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00
30	0.25	0.25	0.25	0.25	0.25	0.25
40	0.75	0.75	0.75	0.75	0.75	0.75
50	1.50	1.50	1.50	1.65	1.40	1.70

3.2 Real Datasets

We have implemented our algorithm on three real benchmark mixed datasets obtained from the UCI Machine Learning repository :
(https://archive.ics.uci.edu/ml/datasets.html).
These datasets are for classification problem (every data point has a class label). Hence, it enables us to evaluate our results based on the consistency of the clustering results with the actual class labels. *We emphasize that, a mismatch between clustering and the actual class labels does not necessarily imply poor performance of the clustering algorithm as the cluster structure may be different from the class structure.* However, we follow this protocol as others have done so. Note that for these datasets, we have performed the Z-score normalization of the numerical attributes.

To evaluate the performance of our algorithms and for comparison with others, we compute the clustering accuracy (r), the number of data points having the same cluster label and class label, after relabeling. We run Algorithm 3 for 30 times for each dataset and computed the mean value of r which we compare with performance of other algorithms. The results for each dataset are summarized next.

Acute Inflammations Dataset This two class dataset consists of 120 data points with each data point having

Table 3: Performance comparison for Acute Inflammations Data (Algorithm 3, $\lambda = 2$)

Algorithm	Clustering Accuracy r
Proposed algorithm	0.883
KL-FCM-GM [6]	0.682 (reported from [20])
Fuzzy k-prototypes [20]	0.710 (reported from [20])
EKP [27]	0.508 (reported from [20])

Table 4: Performance comparison for Heart Disease Data (Algorithm 3, $\lambda = 2$)

Algorithm	Clustering Accuracy r
Proposed algorithm	0.788
Fuzzy k-prototypes [20]	0.835 (reported from [20])
KL-FCM-GM [6]	0.758 (reported from [20])
EKP [27]	0.545 (reported from [20])

Table 5: Performance comparison for Credit Approval Data (Algorithm 3, $\lambda = 10$)

Algorithm	Clustering Accuracy r
Proposed algorithm	0.882
OCIL [7]	0.756 (reported from [7])
Fuzzy k-prototypes [20]	0.838 (reported from [20])
KL-FCM-GM [6]	0.584 (reported from [20])
EKP [27]	0.682 (reported from [20])

one numerical attribute and six categorical attributes. When we use $\lambda = 2$ the clustering quality is very good, it is much better than all three algorithms compared as shown in Table 3. A natural question comes why did we use a higher value of λ? We shall see later (Table 6) that even $\lambda = 10$ produces partitions better than those by all three algorithms compared.

Heart Disease dataset This two-class dataset consists of 303 points with each point having five numerical attributes and eight categorical attributes. Table 4 summarizes the results over 30 iterations of Algorithm 3. Our algorithm performed quite well compared to the existing algorithms, providing better than KL-FCM-GM[6] and EKP[27] algorithms.

Credit Approval Data This dataset consists of 653 data points (after removing data points with missing attributes) distributed in two classes. Each data point is represented by six numerical attributes and nine categorical attributes. In Table 5 we find that our algorithm outperforms all the existing algorithms for this dataset.

Statlog Heart Data This dataset consists of 270 data points distributed in two classes. Each data point is represented by seven numerical attributes and six categorical attributes. In Table 6 we find that our algorithm performs better than two of the existing algorithms, while the performance of the remaining algorithm, OCIL, is marginally better.

Table 6: Performance comparison for Statlog Heart Data (Algorithm 3, $\lambda = 2$)

Algorithm	Clustering Accuracy r
Proposed algorithm	0.822
OCIL [7]	0.824 (reported from [7])
k-Prototypes [15]	0.770 (reported from [7])
k-Means	0.596 (reported from [7])

Table 7: Performance comparison for German Credit Data (Algorithm 3, $\lambda = 1000$)

Algorithm	Clustering Accuracy r
Proposed algorithm	0.609
OCIL [7]	0.694 (reported from [7])
k-Prototypes [15]	0.671 (reported from [7])
k-Means	0.671 (reported from [7])

German Credit Data This dataset consists of 1000 data points distributed in two classes. Each data point is represented by 7 numerical attributes and 13 categorical attributes. In Table 7 we find that our algorithm could not match the performance of the three algorithms for this data sets. To summarize, on average Algorithm 3 is found to work better.

3.2.1 Effect of λ

As we pointed out in Section 2.3, λ is an important parameter for the performance of Algorithm 3. Moreover, the algorithm can be terminated using different conditions such as convergence of $||(U^N + U^C)/2||$, $||U^N||$ and $||U^C||$. In this section, we investigate the effect of the termination condition as well as that of different choices of λ on the performance of Algorithm 3 for the three real datasets. Table 8 shows the summary of the performance of Algorithm 3 for different choices of λ when the termination is done on $||(U^N + U^C)/2||$. In Table 9, we present the same when Algorithm 3 terminated on $||U^N||$. Finally, Table 10 is generated based on termination with $||U^C||$. We emphasize the best result for each dataset in bold. From Table 8, 9, and 10 we find that both λ and the termination criterion have a noticeable effect on the performance of Algorithm 3. In particular, for Heart Disease and Credit Approval, the performance improves as λ increases from 2 to 10, while that is not the case for Acute Inflammation data for which $\lambda = 2$ provides the best result. Comparing Table 9 and Table 10, we find that termination on U^C yields quite good results for Heart Disease and Credit Approval, but this is not the case for Acute Inflammation. The last column, in each of these three tables (Tables

8 - 10) depicts the performance yielded by the strategy of dynamic λ. From these tables, we observe that although the dynamic λ cannot produce the best results, it does produce comparable results. Based on these limited experiments, we may summarize that $\lambda = 100$ is a good choice for which the termination condition does not have much effect across different data sets. Similarly, for dynamic λ the impact of the termination condition is not much for different data sets.Therefore, the strategy of dynamic λ is a useful strategy.

3.2.2 Convergence behavior of the algorithms

For Algorithms 1 and 2, we use alternating optimization to optimize their respective objective function by iterating through the necessary conditions for optimization. Therefore Algorithms 1 and 2 will terminate either at a local minima or at a saddle point. In Fig.3 we show the variation of the objective function for Algorithms 1 and 2 with the number of iterations when run on the Credit Approval dataset. In Fig.4 we show the variation of the cost function for Algorithm 3 with number of iterations when run on the Credit Approval dataset. The figures are for a typical run of the algorithm.

4 Conclusion and Future work

We have proposed a new approach for clustering mixed datasets inspired by the collaborative clustering frameworks. In particular, we have proposed three algorithms. The first algorithm (Algorithm 1) gives more importance to numerical data and the structure found in the categorical data is used a regularizing term, while the second one (Algorithm 2) switches the role of numerical and categorical data; the third one (Algorithm 3), on the other hand, attempts to find cluster structure that is common to both numerical data and categorical data. All three algorithms depend on a regularizing factor λ. An important issue is how to decide on the value of λ. It depends on the datasets and the requirements. For example, if the partition of the numerical data forms the regularizing term, and we want the numerical part to dominate the cluster structure, we need to use a high value of λ. For Algorithm 3, a higher λ, on the other hand, assigns a greater weight for the closeness of the numerical and categorical structures. We have used fixed λ, as well as proposed a strategy that uses dynamic λ. The dynamic λ strategy works fine; however, depending on the dataset, an appropriate choice of fixed λ may work better. This is an issue, that is yet to be resolved.

References

1. Ahmad, A., Dey, L.: A k-mean clustering algorithm for mixed numeric and categorical data. Data & Knowledge Engineering **63**(2), 503–527 (2007)
2. Bezdek, J., Keller, J., Krishnapuram, R., Pal, N.: Fuzzy models and algorithms for pattern recognition and image processing. Norwell, MA (1999)
3. Bezdek, J.C.: Pattern recognition with fuzzy objective function algorithms. Kluwer Academic Publishers (1981)
4. Bezdek, J.C., Ehrlich, R., Full, W.: Fcm: The fuzzy c-means clustering algorithm. Computers & Geosciences **10**(2), 191–203 (1984)
5. Bishop, C.M., et al.: Pattern recognition and machine learning, vol. 4. springer New York (2006)
6. Chatzis, S.P.: A fuzzy c-means-type algorithm for clustering of data with mixed numeric and categorical attributes employing a probabilistic dissimilarity functional. Expert Systems with Applications **38**(7), 8684–8689 (2011)
7. Cheung, Y.m., Jia, H.: Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number. Pattern Recognition **46**(8), 2228–2238 (2013)
8. Coletta, L.F., Vendramin, L., Hruschka, E.R., Campello, R.J., Pedrycz, W.: Collaborative fuzzy clustering algorithms: Some refinements and design guidelines. Fuzzy Systems, IEEE Transactions on **20**(3), 444–462 (2012)
9. Everitt, B.S.: A finite mixture model for the clustering of mixed-mode data. Statistics & probability letters **6**(5), 305–309 (1988)
10. Gath, I., Geva, A.B.: Unsupervised optimal fuzzy clustering. Pattern Analysis and Machine Intelligence, IEEE Transactions on **11**(7), 773–780 (1989)
11. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Applied statistics pp. 100–108 (1979)
12. He, Z., Xu, X., Deng, S.: Squeezer: an efficient algorithm for clustering categorical data. Journal of Computer Science and Technology **17**(5), 611–624 (2002)
13. He, Z., Xu, X., Deng, S.: Clustering mixed numeric and categorical data: A cluster ensemble approach. arXiv preprint cs/0509011 (2005)
14. Honda, K., Ichihashi, H.: Regularized linear fuzzy clustering and probabilistic pca mixture models. Fuzzy Systems, IEEE Transactions on **13**(4), 508–516 (2005)
15. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining,(PAKDD), pp. 21–34. Singapore (1997)
16. Huang, Z.: A fast clustering algorithm to cluster very large categorical data sets in data mining. In: DMKD, p. 0. Citeseer (1997)
17. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery **2**(3), 283–304 (1998)
18. Huang, Z., Ng, M.K.: A fuzzy k-modes algorithm for clustering categorical data. Fuzzy Systems, IEEE Transactions on **7**(4), 446–452 (1999)
19. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM computing surveys (CSUR) **31**(3), 264–323 (1999)
20. Ji, J., Pang, W., Zhou, C., Han, X., Wang, Z.: A fuzzy k-prototype clustering algorithm for mixed numeric and categorical data. Knowledge-Based Systems **30**, 129–135 (2012)

Table 8: Effect of λ, on Mean Accuracy (over 30 runs) and Std. Deviation when $(U^N + U^C)/2$ cluster structure is used

DS	$\lambda = 2$	$\lambda = 10$	$\lambda = 100$	$\lambda = 1000$	λ^*
HD	0.741±0.000	0.757±0.000	0.778±0.000	0.744±0.000	0.778±0.000
CA	0.743±0.000	0.870±0.000	0.851±0.000	0.703±0.000	0.859±0.000
AI	0.850±0.000	0.792±0.000	0.780±0.129	0.747±0.138	0.809±0.069
SH	0.785±0.000	0.785±0.000	0.789±0.000	0.774±0.000	0.789±0.000
GC	0.603±0.000	0.604±0.000	0.608±0.000	0.608±0.001	0.604±0.000

DS: Data Set, HD: Heart Disease, CA: Credit Approval,
AI: Acute Inflammations, SH: Statlog Heart, GC: German Credit

Table 9: Effect of λ, on Mean Accuracy (over 30 runs) and Std. Deviation when U^N cluster structure is used

DS	$\lambda = 2$	$\lambda = 10$	$\lambda = 100$	$\lambda = 1000$	λ^*
HD	0.723±0.000	0.747±0.000	0.774±0.000	0.744±0.000	0.774±0.000
CA†	0.712±0.000	0.827±0.000	0.854±0.001	0.714±0.012	0.856±0.000
AI‡	**0.883**±0.000	0.829±0.004	0.783±0.125	0.747±0.139	0.808±0.063
SH	0.771±0.000	0.778±0.000	0.789±0.000	0.751±0.026	0.789±0.000
GC	0.602±0.000	0.605±0.000	0.605±0.009	0.586±0.046	0.604±0.001

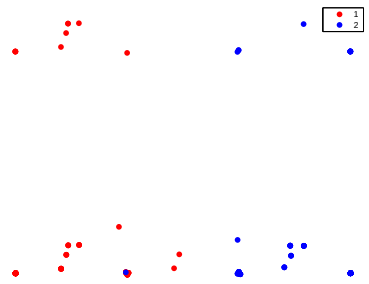
DS: Data Set, HD: Heart Disease, CA: Credit Approval,
AI: Acute Inflammations, SH: Statlog Heart, GC: German Credit

Table 10: Effect of λ, on Mean Accuracy (over 30 runs) and Std. Deviation when U^C cluster structure is used

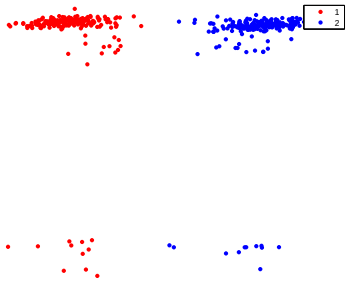
DS	$\lambda = 2$	$\lambda = 10$	$\lambda = 100$	$\lambda = 1000$	λ^*
HD	**0.788**±0.000	0.785±0.000	0.782±0.000	0.744±0.000	0.782±0.000
CA	0.839±0.000	**0.882**±0.000	0.851±0.000	0.703±0.000	0.865±0.000
AI	0.750±0.000	0.750±0.000	0.780±0.129	0.748±0.138	0.809±0.069
SH	**0.822**±0.000	0.781±0.000	0.785±0.000	0.740±0.000	0.785±0.000
GC	0.605±0.000	0.603±0.000	0.607±0.001	**0.609**±0.006	0.605±0.001

DS: Data Set, HD: Heart Disease, CA: Credit Approval,
AI: Acute Inflammations, SH: Statlog Heart, GC: German Credit

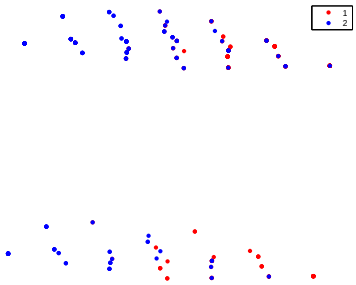
21. Jorgensen, M., Hunt, L.: Mixture model clustering of data sets with categorical and continuous variables. In: Proceedings of the Conference ISIS'96, Australia, pp. 375–84 (1996)
22. Modha, D.S., Spangler, W.S.: Feature weighting in k-means clustering. Mach. Learn. **52**(3), 217–237 (2003). DOI 10.1023/A:1024016609528. URL `http://dx.doi.org/10.1023/A:1024016609528`
23. Pedrycz, W.: Collaborative fuzzy clustering. Pattern Recognition Letters **23**(14), 1675 – 1686 (2002). DOI http://dx.doi.org/10.1016/S0167-8655(02)00130-7. URL `http://www.sciencedirect.com/science/article/pii/S0167865502001307`
24. San, O.M., Huynh, V.N., Nakamori, Y.: An alternative extension of the k-means algorithm for clustering categorical data. International Journal of Applied Mathematics and Computer Science **14**(2), 241–248 (2004)
25. Witold, P., Rai, P.: Collaborative clustering with the use of fuzzy c-means and its quantification. Fuzzy Sets and Systems **159**(18), 2399–2427 (2008)
26. Yang, M.S., Hwang, P.Y., Chen, D.H.: Fuzzy clustering algorithms for mixed feature variables. Fuzzy Sets and Systems **141**(2), 301–317 (2004)
27. Zheng, Z., Gong, M., Ma, J., Jiao, L., Wu, Q.: Unsupervised evolutionary clustering algorithm for mixed type data. In: Evolutionary Computation (CEC), 2010 IEEE Congress on, pp. 1–8. IEEE (2010)



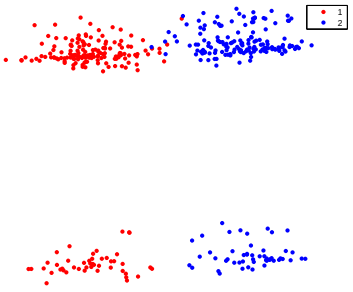
(a) Top two Principal components of the categorical part with 10% noise



(b) Top two Principal components of the entire dataset with 10% noise



(c) Top two Principal components of the categorical part with 50% noise



(d) Top two Principal components of the entire dataset with 50% noise

Fig. 2: Clustering results with different levels of noise on Synth dataset

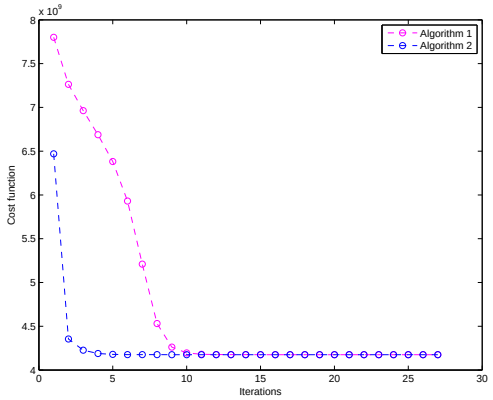


Fig. 3: Convergence plot for Algorithms 1 and 2 for Credit Approval dataset

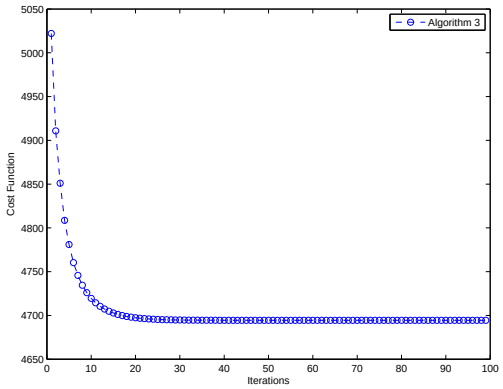


Fig. 4: Convergence plot for Algorithm 3 for Credit Approval dataset