# Clustering Heterogeneous Data with Mutual Semi-Supervision

Artur Abdullin and Olfa Nasraoui

Knowledge Discovery & Web Mining Lab,
Department of Computer Engineering and Computer Science,
University of Louisville, Louisville, KY, USA
{ar.abdullin, olfa.nasraoui}@louisville.edu

**Abstract.** We propose a new methodology for clustering data comprising multiple domains or parts, in such a way that the separate domains mutually supervise each other within a semi-supervised learning framework. Unlike existing uses of semi-supervised learning, our methodology does not assume the presence of labels from part of the data, but rather, each of the different domains of the data separately undergoes an unsupervised learning process, while sending and receiving supervised information in the form of data constraints to/from the other domains. The entire process is an alternation of semi-supervised learning stages on the different data domains, based on Basu et al.'s Hidden Markov Random Fields (HMRF) variation of the K-means algorithm for semi-supervised clustering that combines the constraint-based and distance-based approaches in a unified model. Our experiments demonstrate a successful mutual semi-supervision between the different domains during clustering, that is superior to the traditional heterogeneous domain clustering baselines consisting of converting the domains to a single domain or clustering each of the domains separately.

**Keywords:** mixed data type clustering, heterogeneous data clustering

## 1 Introduction

Recent years have seen the emergence of increasing amounts of heterogeneous or mixed-type data that consists of several parts, each part being a different type of domain or modality, for example many Web data sets, network activity data (e.g. the KDD cup data), scientific data sets, and demographic and census data sets typically comprise several parts that are of different types: numerical, categorical, transactional, free text, ratings, social relationships, etc. Traditionally each of these different types of data has been best clustered with a different specialized clustering algorithm or with a specialized dissimilarity measure. A very common approach to cluster data with mixed types has been to either convert all data types to the same type (e.g: from categorical to numerical or vice-versa) and then cluster the data with a standard clustering algorithm that is suitable for that target domain; or to use a different dissimilarity measure for each domain,

then combine them into one dissimilarity measure and cluster this dissimilarity matrix with an $O(N^2)$ algorithm.

In this paper, we investigate a new methodology to handle heterogeneous data consisting of different or mixed data types. Similar to our preliminary work in [1], our approach makes an innovative use of Semi-Supervised Learning (SSL), which is used in a completely novel way and for a new purpose that has never been the objective of previous SSL research and applications. Unlike our preliminary work [1] that relied on the exchange of seeds as the semi-supervising link between the alternating clustering processes of the different data types or domains, in this paper we use cluster-membership constraints as the semi-supervising link between the processes.

Whereas traditional semi-supervised learning or transductive learning has been used mainly to exploit additional information in unlabeled data to enhance the performance of a classification model trained with labeled data [4], or to exploit external *supervision* in the form of some labeled data to enhance the results of clustering unlabeled data; the methodology presented in this paper uses SSL "without" any external labels. In fact, the guiding or semi-supervising labels will be "inferred" from multiple Semi-supervised Learners (SSL), such that each SSL transmits to the other SSL, a subset of confident pairwise must-link constraints (MLC) and cannot-link constraints (CLC) that it has learned on its own from the data in one domain, and that try to favor placing some data records (in the MLC) in the same cluster while trying to forbid others (in the CLC) from being placed in the same cluster. Hence the SSLs from the different domains try to mutually guide each other with each separate SSL transmitting semi-supervision constraints to the other SSL in the other domain, according to what it has discovered in its own domain. For the SSL, we chose Basu et al.'s Hidden Markov Random Fields (HMRF) K-means algorithm that combines the constraint-based and distance-based approaches in a unified model. In addition to this method being a principled and rigorous approach, our choice was motivated by a review of many SSL algorithms and by the comparative experimental results in [4] that reported the superiority of this approach over many others. Last by not least, our choice was also motivated by the flexibility of Basu's approach that allows the freedom of customizing many optional components of the semi-supervision from weighting the different constraints to learning the distance measure. This in turn makes the approach more open to extensions and investigating different options in our methodology. Moreover, as we will show in this paper, using the HMRF K-means as a basis within our framework outperformed the seed exchange-based SSL framework presented in our preliminary work in [1].

The rest of this paper is organized as follows. Section 2 gives an overview of related work. Section 3 presents our proposed framework to cluster mixed data. Section 4 evaluates the proposed approach and Section 5 presents our conclusions.

## 2 Related Work

Most well known clustering algorithms are specialized for specific types of attributes. For instance, categorical attributes have been handled using specialized algorithms such as k-modes, ROCK or CACTUS [12,11,9]. The spherical k-means algorithm is a variant of the k-means algorithm that uses the cosine similarity instead of the Euclidean distance. The algorithm computes a disjoint partition of the document vectors and for each cluster, computes a centroid that is then normalized to have unit Euclidean norm [6]. This algorithm was successfully used for clustering text documents which are often represented as sparse high-dimensional vector data. Numerical data has been clustered using k-means, DBSCAN and many other algorithms [14,7].

The above approaches have the following limitations:

- Specialized clustering algorithms can fall short when they must handle different data types.
- The alternative of data conversion to a single type can result in the loss of information or the creation of artifacts in the data.
- In the case where different parts or domains of the data originate from multiple sources, they may be hard to combine for the purpose of clustering because of the problem of duplication of data and the problem of missing data from one of the sources, in addition to the problem of heterogeneous types of data from multiple sources.

Algorithms for mixed data attributes exist, for instance the k-prototypes [13] and INCONCO algorithms [15]. The k-prototypes algorithm integrates the k-means and the k-modes algorithms to allow for clustering objects described by mixed numerical and categorical attributes, by combining their (weighted) contributions to a distance measure. However it is limited by the fact that the choice of the weighting parameters cannot vary from one cluster to another, and is critical to the clustering success. The INCONCO algorithm extends the Cholesky decomposition to model dependencies in heterogeneous data and, relying on the principle of Minimum Description Length, integrates numerical and categorical information in clustering. The limitations of INCONCO include that it assumes a known probability distribution model for each domain, and it assumes that the number of clusters is identical in both domains.

### 2.1 Ensemble-based Clustering

Another direction that has been attracting growing interest in machine learning is *ensemble learning*, in particular *ensemble-based clustering* for the unsupervised learning task [2,10]. Ensemble-based clustering methods typically aim to combine the *end* results of several clustering runs or algorithms, where the runs can be on the same or different parts of the data. Our proposed approach is reminiscent of ensemble-based clustering. However, one main distinction is that our approach enables the different algorithms running in each domain to reinforce or supervise each other during the *intermediate* stages, until the final clustering is obtained. In other words, our approach is more collaborative.

### 2.2 Semi-supervised Clustering

Apart from clustering algorithms, which are unsupervised learners in the sense that they use *unlabeled* data, recent years have seen increasing interest in another direction, known as *semi-supervised learning* which takes advantage from both labeled and unlabeled data. Many semi-supervised algorithms have been proposed including co-training, transductive support vector machines, entropy minimization, semi-supervised Expectation Maximization , graph-based approaches, and clustering-based approaches. In semi-supervised clustering, labeled data can be used in the form of *(1) initial seeds* [3], *(2) constraints* [18], or *(3) feedback* [5]. All these existing approaches are based on model-based clustering where each cluster is represented by its centroid. *Seed-based* approaches use labeled data *only to help initialize* cluster centroids, while *constrained* approaches keep the grouping of labeled data unchanged throughout the clustering process, and *feedback-based* approaches start by running a regular clustering process and finally adjusting the resulting clusters based on labeled data.

**Semi-supervised Clustering with HMRF-KMeans** The HMRF-KMeans algorithm [4] provides a principled probabilistic framework for incorporating supervision into prototype based clustering by using an objective function that is derived from the posterior energy of the Hidden Markov Random Fields framework for the constrained cluster label assignments. The HMRF consists of the hidden field of random variables with unobservable values corresponding to the cluster assignments/labels of the data, and an observable set of random variables which are the input data. The neighborhood structure over the hidden labels is defined based on the constraints between data point assignments (the neighbors of a data point are the points that are related to it via must-link or cannot-link constraints). The HMRF-KMeans algorithm is an Expectation Maximization (EM) based partitional clustering algorithm for semi-supervised clustering that combines the constraint-based and distance-based approaches in a unified model. First, let us introduce the pertinent notation: $X$ refer to a set of objects, whose representatives are enumerated as $\{x_i\}_{i=1}^{N}$, $x_{im}$ represents the $m^{th}$ component of the $d$-dimensional vector $x_i$. This semi-supervised clustering model accepts as input a set of data points $X$ with a specified distortion measure $D$ between the points, and external supervision that is provided by a set of must-link constraints $M = \{(x_i, x_j)\}$ (with its set of associated violation costs $W$) and a set of cannot-link constraints $C = \{(x_i, x_j)\}$ (with its associated violation costs $\bar{W}$). The goal of the algorithm is to partition the data into $K$ clusters so that the total of the distortions $D$ between the points and their corresponding cluster representatives $\{\mu_h\}_{h=1}^{K}$ is minimized while violating a minimum number of constraints. The HMRF-KMeans objective function in (1) consists of four terms. The first term sums the distances between data objects and their corresponding cluster representatives. The second term adds a must-link violation penalty, which penalizes distant points that violate the must-link constraint higher compared to nearby points. This has the effect of penalizing the objective function to bring a pair of points that violate a must-link constraint closer to each other.

Analogously, the next term represents the penalties for violating cannot-link constraints between pairs of data points thus encouraging the distance learning step to put cannot-linked points farther apart. Finally, the last term represents a normalization constant. The objective function [4] is given by

$$J_{obj} = \sum_{x_i \in X} D(x_i, \mu_{l_i}) + \sum_{(x_i,x_j)\in M} w_{ij}\phi_D(x_i, x_j)I[l_i \neq l_j] + \sum_{(x_i,x_j)\in C} \bar{w_{ij}}(\phi_{D_{max}} - \phi_D(x_i, x_j))I[l_i = l_j] + \log Z, \quad (1)$$

where $D(x_i, \mu_{l_i})$ is the distortion between $x_i$ and $\mu_{l_i}$, $w_{ij}$ is the cost of violating the must-link constraint $(i, j)$, $\phi_D(x_i, x_j)$ is the penalty scaling function, chosen to be a monotonically increasing function of the distance between $x_i$ and $x_j$ according to the current distortion measure $D$. $I$ is the indicator function ($I(true) = 1$, $I(false) = 0$), so that the must-link term is active only when cluster labels of $x_i$ and $x_j$ are different. In the next term, $\bar{w_{ij}}$ is the cost of violating the cannot-link constraint $(i, j)$, $\phi_{D_{max}}$ is the maximum value of the scaling function $\phi_D$ for the data set, and $Z$ is a normalization constant. Thus, the task is to minimize $J_{obj}$ over cluster representatives $\{\mu_h\}_{h=1}^{K}$, cluster label configuration $L = \{l_i\}_{i=1}^{N}$ (every $l_i$ takes values from the set $\{1, ..., K\}$), and $D$ (if the distortion measure is parameterized). Many distortion measures can be parameterized [19] and integrated into the HMRF-KMeans algorithm. In this work, we do not parametrize any distortion measure, and instead keep it as a function only of the data objects $D = D(x_i, x_j)$.

The main idea of HMRF-KMeans is as follows: in the E-step, given the current cluster representatives, every data point is re-assigned to the cluster that minimizes its contribution to $J_{ojb}$. In the M-step, the cluster representatives $\{\mu_h\}_{h=1}^{N}$ are re-estimated from the previous cluster assignments to minimize $J_{obj}$ for the current assignment. The E-step and M-step are repeatedly alternated till a specified convergence criterion is reached.

## 3 Proposed Mutual Semi-Supervision Based Heterogeneous Data Clustering using HMRF-KMeans

The HMRF-KMeans algorithm is flexible in the choice of the distortion measure $D$, however a single distortion measure must be used since the data is supposed to be of the same type or domain. In contrast, our data records consist of different domains, thus we will invoke several HMRF-KMeans processes one per domain, with each one receiving supervising constraints that were discovered in the other domains . For the sake of simplicity, we shall limit the data to consist of two parts in the rest of this paper: numerical and categorical. We start by dividing the set of attributes into two subsets: one subset, called domain $T_1$, with only attributes of one type, say numerical, such as $T_1 = \{$age, income, ..., etc$\}$, and a second subset, called $T_2$, with attributes of the other (say categorical) type such as $T_2 = \{$eye color, gender, ..., etc$\}$. The first subset consists of $d_{T_1}$ attributes

from domain $T_1$ and the second subset consists of $d_{T_2}$ attributes from domain $T_2$, such that that $d_{T_1} + d_{T_2} = d$, the total number of dimensions in the data. We use the Euclidean distance and simple matching distance $\delta$ as a distortion measure $D$ for the numerical and categorical domains, respectively. We also define the penalty scaling function $\phi_D(x_i, x_j)$ to be equal to the corresponding distance function, and set the pairwise constraint violation costs $W$ and $\bar{W}$ to unit costs, so that $w_{ij} = \bar{w_{ij}} = 1$ for any pair $(i, j)$.

Putting all this into (1) gives the following objective functions for the numerical domain $T_1$, with $x_{im}$ denoting the $m^{th}$ attribute of data record $x_i$,

$$J_{T_1} = \sum_{x_i \in X} \sqrt{\sum_{m \in T_1} (x_{im} - \mu_{l_i m})^2} + \sum_{(x_i, x_j) \in M_{T_2}} \sqrt{\sum_{m \in T_1} (x_{im} - x_{jm})^2} I[l_i \neq l_j] + \sum_{(x_i, x_j) \in C_{T_2}} (\phi_{D_{T_1, max}} - \sqrt{\sum_{m \in T_1} (x_{im} - x_{jm})^2}) I[l_i = l_j] + \log Z_{T_1}, \quad (2)$$

and for the categorical domain $T_2$:

$$J_{T_2} = \sum_{x_i \in X} \sum_{m \in T_2} \delta(x_{im}, \mu_{l_i m}) + \sum_{(x_i, x_j) \in M_{T_1}} \sum_{m \in T_2} \delta(x_{im}, x_{jm}) I[l_i \neq l_j] + \sum_{(x_i, x_j) \in C_{T_1}} (d_{T_2} - \sum_{m \in T_2} \delta(x_{im}, x_{jm})) I[l_i = l_j] + \log Z_{T_2}. \quad (3)$$

where $M_{T_r}$ is a set of must-link, and $C_{T_r}$ is a set of cannot-link constraints inferred based on the clustering of domain $T_r$. We further set the normalization constants $Z_{T_1}$ and $Z_{T_2}$ to be constant throughout the clustering iterations, and hence drop these terms from Equations 2 and 3.

In our initial work [1], the number of clusters was assumed to be the same in each domain. This can be considered as the default approach, and has the advantage of being easier to design. However, in real life data, the different domains can have different numbers of clusters. One advantage of the constraint-based supervision, used in the new methodology presented in this paper, is that it naturally solves the problem of clustering domains with different numbers of clusters.

### 3.1 Algorithm Flow

Our initial implementation, reported in this paper, can handle data records composed of two parts (such as numerical and categorical) within a semi-supervised framework that consists of the following stages:

1. The first stage consists of dividing the set of attributes into two subsets: one subset, called domain $T_1$, with only attributes of one type, e.g. numerical, (age, income, etc), and another subset, called domain $T_2$, with attributes of another type, e.g. categorical (eyes color, gender, etc).

2. The next stage is to cluster one of the subsets $T_1$ or $T_2$ with the HMRF-KMeans algorithm without any constraints. Ideally, we try to start from the most promising domain in terms of data quality and guiding the clustering process, let us for simplicity assume that we start with domain $T_1$. The HMRF-KMeans algorithm runs for a small number of iterations $t_{T_1}$ and yields a set of $K_{T_1}$ cluster representatives $\{\mu_h\}_{h=1}^{K_{T_1}}$ in that domain by minimizing Equation 2 with no constraints coming from the other domain, i.e. $C_{T_2} = M_{T_2} = \emptyset$.
3. In the third stage, for each of the $K_{T_1}$ cluster representatives $\mu_h$ we find the $n_{T_1}$ closest points, according to the corresponding distance measure in domain $T_1$. Then using those $K_{T_1} \times n_{T_1}$ points, we generate pairwise must-link constraints $M_{T_1}$ using points that belong to the same cluster, and cannot-link constraints $C_{T_1}$ using points that belong to different clusters. These constraints will later be sent to the clustering process in the other domain ($T_2$) in the next stage.
4. In this stage, we cluster data in domain $T_2$ with the HMRF-KMeans algorithm using the entire objective function penalized via the must-link constraints $M_{T_1}$ and cannot-link constraints $C_{T_1}$ obtained from the domain clustered in the previous stage. The HMRF-KMeans algorithm runs for a small number of iterations $t_{T_2}$ and yields a set of cluster representatives $\{\mu_h\}_{h=1}^{K_{T_2}}$ by minimizing Equation 3. Then again, for each cluster representative $\mu_h$ we find the $n_{T_2}$ closest points, according to the corresponding distance measure in domain $T_2$, and generate must-link constraints $M_{T_2}$ and cannot-link constraints $C_{T_2}$ using those points (as explained in detail in stage 3).
5. Similarly, in the next stage, we use the previous domain's must-link constraints $M_{T_2}$ and cannot-link constraints $C_{T_2}$ obtained from stage 4 to penalize the objective function (2) in the HMRF-KMeans algorithm which runs for $t_{T_1}$ iterations and yields a set of cluster representatives $\{\mu_h\}_{h=1}^{K_{T_1}}$ by minimizing Equation 2. Then, for each cluster representative $\mu_h$, we recompute the $n_{T_1}$ closest points, and generate must-link constraints $M_{T_1}$ and cannot-link constraints $C_{T_1}$ using those points.

We repeat stages 4 and 5 until both algorithms converge or the number of exchange iterations exceeds a maximum number. The general flow of our approach is presented in Figure 1. We compared the proposed mixed-type clustering approach with the following two classical baseline approaches for clustering mixed numerical and categorical data.

*Baseline 1: Conversion:* The first baseline approach is to convert all data to the same attribute type and cluster it. We call this method the *conversion* algorithm. Since we have attributes of two types, there are two options to perform this algorithm:

1. Convert all numerical type attributes to categorical type attributes and run k-modes.
2. Convert all categorical type attributes to numerical type attributes and run k-means.

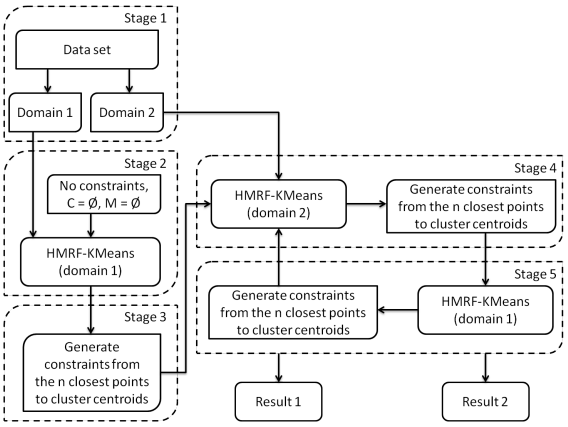


**Fig. 1.** Outline of the mutual semi-supervision based heterogeneous data clustering using HMRF-KMeans.

The conversion algorithm requires data type conversion: from numerical to categorical and from categorical to numerical. There are several ways to convert a numerical type attribute $z$, ranging in $[z_{min}, z_{max}]$, to a categorical type attribute $y$, also known as "discretization". In the current implementation, we use cluster-based conversion which starts by clustering the $n$ numerical values into $N$ clusters using any numerical clustering algorithm (e.g. k-means). The optimal number of clusters $N$ is chosen based on the Silhouette index. We convert categorical type attributes to numerical type attributes by mapping the $n$ values of a nominal attribute to binary values using 1-of-$n$ encoding, resulting into transactional-like data, with each nominal value becoming a distinct binary attribute.

*Baseline 2: Splitting:* The second classical baseline approach is to run k-means and k-modes independently on the numerical and categorical subsets of attributes, respectively. We call this method the *splitting* algorithm.

### 3.2 Computational Complexity

The complexity of the proposed approach is mainly determined by the HMRF-KMeans algorithm, which incurs the heaviest cost during the initialization stage that uses both types of constraints and the unlabeled data to first compute the transitive closure on the must-link constraints to get connected components $\lambda$, consisting of points connected by must-link constraints [4], a procedure that costs $O(N^3)$ time and $O(N^2)$ space. Then for each pair of connected components with at least one cannot-link constraint between them, we add cannot-link constraints between every pair of points in that pair of connected components. This operation takes $O(\lambda^2)$ time, thus $O(K^2)$ of time, since $\lambda$ is an order of $K$.

The second stage of the initialization is the cluster selection which is $O(K^2)$. The initialization step in the HMRF-KMeans is optional but essential for the success of the partitional clustering algorithm. The EM-based minimization of the HMRF-KMeans algorithm is $O(N)$. Finally, we need to account for the overhead complexity resulting from the process of coordination of and alternation of the constraint exchanges between the different domains during the mutual supervision process. This process finds the $K \times n_T$ closest points to the cluster representatives in time $O(N)$ for each domain, then generates the pairwise must-link and cannot-link constraints using those points in constant time. Thus the total computational complexity of the proposed approach is $O(N)^3$ or $O(N)$, depending on whether we perform the initialization step or not, respectively.

## 4 Experimental Results

### 4.1 Clustering Evaluation

The proposed semi-supervised framework was evaluated using internal and external cluster validity metrics. As an internal evaluation measure we used the Silhouette index, which is calculated based on the average silhouette width for each sample, average silhouette width for each cluster and overall silhouette width for the entire data set [16]. Note that calculating the Silhouette index requires a distance measure, thus we used the square of the Euclidean distance for numerical data types and the simple matching distance for categorical data types. We also used the Normalized Mutual Information (NMI) as an external evaluation measure which estimates the quality of the clustering with respect to a groundtruth class membership [17]. NMI measures how closely a clustering algorithm could reconstruct the underlying label distribution in the data.

### 4.2 Real Data Sets

We experimented with three real-life data sets with the characteristics shown in Table 1. All three data sets were obtained from the UCI Machine Learning Repository [8].

**Table 1.** Data sets properties

| Data set | No. of Records | No. of Numerical Attributes | No. of Categorical Attributes | Missing Values | No. of Classes |
|---|---|---|---|---|---|
| Adult | 45179 | 6 | 8 | Yes | 2 |
| Heart Disease Data | 303 | 6 | 7 | Yes | 2 |
| Credit Approval Data | 690 | 6 | 9 | Yes | 2 |

– *Adult Data*. The adult data set was extracted by Barry Becker from the 1994 Census database. The data set has two classes: People who make over \$50K a year and people who make less than \$50K. The original data set consists of $48,842$ instances. After deleting instances with missing and duplicate attributes we obtained $45,179$ instances.

- *Heart Disease Data.* The heart disease data, generated at the Cleveland Clinic, contains a mixture of categorical and numerical features. The data comes from two classes: people with no heart disease and people with different degrees of heart disease.
- *Credit Approval Data.* The data set has 690 instances, which were classified in two classes: approved and rejected.

### 4.3 Results with the Real Data Sets.

Since all three data sets have two classes, we clustered them in two clusters[1]. We repeated each experiment 50 times (10 times for the larger adult data set), and report the mean, standard deviation, minimum, median, and maximum values for each validation metric (in the format of mean±std [min, median, max]). Table 2 shows the results of the real data set using the proposed mutual semi-supervision based heterogeneous data clustering framework using HMRF-KMeans, the conversion algorithm, and the splitting algorithm, with the best results in a bold font, based on significant p-values. The results are described below for each data set.

- Adult Data: As Table 2 illustrates, the proposed method performs better in both domains: showing significant improvements in the Silhouette index for the numerical domain and both validity indices for the categorical domain. Note the high maximum value of the Silhouette index in the numerical domain, showing that over many runs, the proposed approach is able to achieve a better top clustering result than classical baseline approaches.
- Heart Disease Data: The conversion algorithms yielded better clustering results for the numerical domain based on the NMI index, however the proposed approach outperformed the conversion algorithm in the categorical domain while conceding to the splitting algorithm.
- Credit Approval Data: The proposed approach outperforms the traditional algorithms for the categorical type attributes based on both internal and external indices. It also performs better in the numerical domain in terms of NMI but concedes to the splitting algorithm in terms of the Silhouette index. One possible reason is that the cluster structure does not match the "true" class labels or ground truth, which is common in unsupervised learning.

## 5 Conclusions

Our preliminary results show that the proposed proposed mutual semi-supervision based heterogeneous data clustering framework using HMRF-KMeans tends to yield better clustering results in the categorical domain. Thus the constraints obtained from clustering the numerical domain tend to provide additional helpful

[1] we realize the possibility of more clusters per class but defer this to future experiments

**Table 2.** Clustering result for the real data sets.

| Data set | Adult | | |
|---|---|---|---|
| Data type | Numerical | | |
| Algorithm | Semi-supervised | Conversion | Splitting |
| Silhouette Index | **0.92**±0.00[0.92, 0.92, 0.92] | $0.07 \pm 0.05[-0.02, 0.08, 0.17]$ | $0.21 \pm 0.0[0.21, 0.21, 0.21]$ |
| NMI | $0.06 \pm 0.00[0.06, 0.06, 0.06]$ | $0.08 \pm 0.07[2.1e-4, 0.13, 0.13]$ | 0.10±0.00[0.10, 0.10, 0.10] |
| Data type | Categorical | | |
| Algorithm | Semi-supervised | Conversion | Splitting |
| Silhouette Index | **0.28**±0.00[0.28, 0.28, 0.28] | $0.22 \pm 0.02[0.19, 0.21, 0.25]$ | $0.25 \pm 0.01[0.24, 0.24, 0.27]$ |
| NMI | **0.17**±0.00[0.17, 0.17, 0.17] | $0.09 \pm 0.02[0.07, 0.08, 0.12]$ | $0.09 \pm 0.01[0.08, 0.08, 0.11]$ |
| Data set | Heart disease | | |
| Data type | Numerical | | |
| Algorithm | Semi-supervised | Conversion | Splitting |
| Silhouette Index | **0.36**±0.00[0.18, 0.21, 0.71] | $0.26 \pm 0.07[0.16, 0.18, 0.19]$ | $0.36 \pm 0.00[0.36, 0.36, 0.36]$ |
| NMI | $0.24 \pm 0.00[0.24, 0.24, 0.24]$ | $\mathbf{0.28} \pm 0.11[2.1e-4, 0.32, 0.32]$ | 0.19±0.00[0.18, 0.19, 0.19] |
| Date type | Categorical | | |
| Algorithm | Semi-supervised | Conversion | Splitting |
| Silhouette Index | 0.29±0.00[0.29, 0.29, 0.29] | $0.18 \pm 0.00[0.16, 0.18, 0.19]$ | $\mathbf{0.31} \pm 0.01[0.29, 0.31, 0.31]$ |
| NMI | 0.27±0.00[0.27, 0.27, 0.27] | $0.26 \pm 0.00[0.18, 0.25, 0.36]$ | $\mathbf{0.29} \pm 0.02[0.23, 0.30, 0.30]$ |
| Data set | Credit card | | |
| Data type | Numerical | | |
| Algorithm | Semi-supervised | Conversion | Splitting |
| Silhouette Index | $0.46 \pm 0.01[0.45, 0.46, 0.46]$ | $0.35 \pm 0.27[0.12, 0.29, 0.92]$ | $\mathbf{0.63} \pm 0.06[0.62, 0.62, 0.95]$ |
| NMI | $\mathbf{0.20} \pm 0.00[0.19, 0.20, 0.20]$ | $0.13 \pm 0.13[2.1e-4, 0.03, 0.31]$ | 0.08±0.01[0.03, 0.08, 0.08] |
| Date type | Categorical | | |
| Algorithm | Semi-supervised | Conversion | Splitting |
| Silhouette Index | $\mathbf{0.23} \pm 0.01[0.22, 0.22, 023]$ | $0.17 \pm 0.02[0.13, 0.16, 0.21]$ | $0.23 \pm 0.01[0.19, 0.23, 0.24]$ |
| NMI | $\mathbf{0.28} \pm 0.01[0.27, 0.28, 0.28]$ | $0.23 \pm 0.03[0.12, 0.23, 0.31]$ | $0.26 \pm 0.02[0.22, 0.27, 0.36]$ |

knowledge to the categorical clustering algorithm. This information may in turn be used to avoid local minima and obtain a better clustering in the categorical domain. We are currently completing our study by extending our experiments and methodology to mixed data involving transactional information (particularly text and clickstreams) as one of the data domains. In the future, we plan to investigate the effect of parameterized distortion measures that are incorporated with the proposed heterogeneous data clustering framework. We also plan to devise a better method to estimate the confidence levels of the points contributing to the created constraints, and then use them to obtain better informed constraint violation cost weights $W$ and $\bar{W}$. We are also extending our experiments to study the sensitivity of the proposed framework to its parameters, such as $n_T$, the number of closest points to the cluster representatives, and the number of iterations $t_T$ when running the HMRF-KMeans algorithm in each stage. Last but not least, we are investigating ways to reduce the cost of the initialization step.

## Acknowledgment

This work was supported by US National Science Foundation *Data Intensive Computation* Grant IIS-0916489.

## References


1. Abdullin, A., Nasraoui, O.: A semi-supervised learning framework to cluster mixed data types. In: Proceedings of KDIR 2012 - International Conference on Knowledge Discovery and Information Retrieval (2012)
2. Al-Razgan, M., Domeniconi, C.: Weighted clustering ensembles. In: Proc. of the 6th SIAM ICML (2006)
3. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: Proc. of 19th ICML (2002)
4. Basu, S., Bilenko, M., Mooney, R.J.: A probabilistic framework for semi-supervised clustering. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 59–68. KDD '04 (2004)
5. Cohn, D., Caruana, R., Mccallum, A.: Semi-supervised clustering with user feedback. Tech. rep. (2003)
6. Dhillon, I.S., Modha, D.S.: Concept decompositions for large sparse text data using clustering. Mach. Learn. 42, 143–175 (2001)
7. Ester, M., peter Kriegel, H., S, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the Second International Conference on KDD. pp. 226–231 (1996)
8. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
9. Ganti, V., Gehrke, J., Ramakrishnan, R.: Cactus - clustering categorical data using summaries. In: Proc. of the 5th ACM SIGKDD International Conference on KDD. pp. 73–83 (1999)
10. Ghaemi, R., Sulaiman, M.N., Ibrahim, H., Mustapha, N.: A survey: Clustering ensembles techniques (2009)
11. Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes. Information Systems 25, 345 – 366 (2000)
12. Huang, Z.: A fast clustering algorithm to cluster very large categorical data sets in data mining. In: In Research Issues on KDD. pp. 1–8 (1997)
13. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery 2, 283–304 (1998)
14. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proc. of the 5th Berkeley Symposium on Math. Statistics and Probability. vol. 1, pp. 281–297 (1967)
15. Plant, C., Böhm, C.: Inconco: interpretable clustering of numerical and categorical objects. In: Proc. of the 17th ACM SIGKDD International Conference on KDD. pp. 1127–1135 (2011)
16. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65 (1987)
17. Strehl, A., Strehl, E., Ghosh, J., Mooney, R.: Impact of similarity measures on web-page clustering. In: Workshop on AI for Web Search. pp. 58–64 (2000)
18. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Proc. of the 18th ICML. pp. 577–584 (2001)
19. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side-information. In: Advances in Neural Information Processing Systems 15. pp. 505–512. MIT Press (2002)