

**Clustering Mixed-Type Data with Correlation-Preserving Embedding**

[](http://crossmark.crossref.org/dialog/?doi=10.1007/978-3-030-73197-7_23&domain=pdf)

Luan Tran1(B) , Liyue Fan2 , and Cyrus Shahabi1

1 University of Southern California, Los Angeles, USA {luantran,shahabi}@usc .edu

2 University of North Carolina at Charlotte, Charlotte, USA liyue .fan@uncc .edu

**Abstract.** Mixed-type data that contains both categorical and numer- ical features is prevalent in many real-world applications. Clustering mixed-type data is challenging, especially because of the complex rela- tionship between categorical and numerical features. Unfortunately, widely adopted encoding methods and existing representation learning algorithms fail to capture these complex relationships. In this paper, we propose a new correlation-preserving embedding framework, COPE, to learn the representation of categorical features in mixed-type data while preserving the correlation between numerical and categorical features. Our extensive experiments with real-world datasets show that COPE generates high-quality representations and outperforms the state-of-the- art clustering algorithms by a wide margin.

**Keywords:** Mixed-type data · Clustering · Correlation preserving

**1 Introduction**

Mixed-type data, which contains both categorical and numerical features, is ubiquitous in the real world. It appears in many domains such as in network data [[34](#bookmark1)] with the size of packages (numerical) and protocol type (categorical),and in personal data [[26](#bookmark2)] with gender (categorical) and income information (numerical). Clustering is an important data mining task that groups data objects into clus- ters so that the objects in the same cluster are more similar to each other than to those in other clusters. Mixed-type data clustering has many real-world appli- cations such as customer segmentation for diﬀerentiated targeting in marketing [[17](#bookmark3)] and health data analysis [[38](#bookmark4)[,40](#bookmark5)]. However, most of the existing clustering algorithms have been developed for only numerical [[2](#bookmark6)[,12](#bookmark7),[14](#bookmark8),[18](#bookmark9),[37](#bookmark10)] or categorical data [[1](#bookmark11),[4](#bookmark12),[10](#bookmark13),[22](#bookmark14),[31](#bookmark15)]. There are only a handful of algorithms [[6](#bookmark16),[16](#bookmark17),[22](#bookmark14),[30](#bookmark18)] designed for mixed-type data.

A common approach to cluster mixed-type data is to generate numerical representations for categorical features, e.g., by using onehot encoding, then apply clustering algorithms designed for numerical data. The challenge of this approach is ﬁnding a good numerical representation that captures the complex oc Springer Nature Switzerland AG 2021

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12682, pp. 342–358, 2021 .

<https://doi.org/10.1007/978-3-030-73197-7_23>

Clustering Mixed-Type Data with Correlation-Preserving Embedding 343

relationship between numerical and categorical features. Simple encoding meth- ods, such as onehot, ordinal, and binary encoding, operate on individual features separately, and do not consider the relationship between features. In recent years, neural networks [[27](#bookmark19)[,35](#bookmark20)] have become a popular choice for representation learning because of its capacity in approximating complex functions. Autoencoder [[35](#bookmark20)] is a natural choice for using a neural network to learn the data representation. It is a typical neural model with full connections between features and hidden units. However, a simple autoencoder that minimizes the reconstruction error loss may not fully capture the correlation between features.

This paper proposes a COrrelation-Preserving Embedding framework (COPE) to learn the representation for categorical features while preserving the relationship between categorical and numerical features. The COPE frame- work improves the representation learned by an autoencoder by incorporating two sub-networks to capture the correlation between categorical, numerical, and embedded data. By concurrently optimizing for representation learning and cor- relation preservation, the embedded categorical data preserves its semantics and the relationship with numerical features, thus providing more accurate clustering results.

We evaluate our proposed approach using six real-world datasets in various domains. Our extensive experimental results show that COPE outperforms other methods in clustering metrics such as Adjusted Mutual Information (AMI) [[36](#bookmark21)] and Fowlkes-Mallows Index (FMI) [[15](#bookmark22)]. The qualitative representation analysis using t-SNE visualization [[28](#bookmark23)] depicts the eﬀectiveness of COPE in grouping similar data into clusters. The convergence test shows that the COPE network quickly converges after a few iterations.

The remainder of this paper is organized as follows. In Sect.[2](#bookmark24), we formally deﬁne the mixed-type data clustering problem and provide an overview of the correlation between categorical and numerical features. In Sect.[3](#bookmark25), we discuss the current approaches for mixed-type data clustering. In Sect.[4](#bookmark26), we introduce our proposed approach COPE. In Sect.[5](#bookmark27), we present our experimental results in detail. We conclude the paper with discussion and future research directions in Sect.[6.](#bookmark28)

**2 Background**

**2.1 Problem Deﬁnition**

Let us denote X = {x1 , x2 , ..., xN } ∈ X as a set of N objects in which each object has dc categorical features and dn continuous features,i.e., F = Fc ∪Fn

where Fc = {f, ..., fc } and Fn = {f, ..., fn }. Each categorical feature fi

has adiscrete value domain Vi = {v, v , ...}. For each data object x, its value

in a continuous and categorical feature is denoted by xn ∈ Xn and xc ∈ Xc , respectively. The problem can be deﬁned as ﬁnding a good representation of data points, which preserves the complex relationship between the features to cluster data points accurately.

344 L. Tran et al.

**Table 1.** Example mixed-type data

|  |  |  |
| --- | --- | --- |
| Area | Shape | Color |
| 10 | Triangle | Blue |
| 12 | Triangle | Blue |
| 30 | Circle | Red |
| 50 | Circle | Red |
| 45 | Diamond | Red |
| 28 | Diamond | Red |
| 8 | Square | Blue |
| 7 | Square | Blue |

The challenge is that there might not be an order or apparent distances between categorical values; hence, it is impossible to compute a co-variance matrix of numerical and categorical features. We need to infer the relation- ship between categorical and numerical features from the data. Table [1](#bookmark29) demon- strates one example of mixed-type data. Each row presents three features of one object. The features “Shape” and “Color” are categorical, and the feature “Area” is numerical. We assume that the order between the values in the cat- egorical features is not known a priori. From the data, we can infer that an object with blue color and a triangle or square shape tends to have a small area (less than 20). Therefore, a triangle can be inferred to be more “similar” to a square than a circle. However, simple encoding methods, e.g., ordinal encoding: { triangle → 0, circle → 1, diamond → 2, and square → 3 }, do not capture that relationship. To solve this problem, we need a mechanism to measure and

preserve the correlation between categorical and numerical features.

**2.2 Correlation Between Numerical and Categorical Data**

There are two main approaches to measure the correlation between a numerical and a categorical feature,i.e., point biserial correlation [[33](#bookmark30)],and regression [[21](#bookmark31)]. The point biserial correlation coeﬃcient is a special case of Pearson’s correla- tion coeﬃcient [[7](#bookmark32)]. It assumes the numerical variables are normally distributed and homoscedastic. The point biserial correlation is from −1 to 1. In the sec- ond approach, the intuition is that if there is a relationship between categorical and numerical features, we should be able to construct an accurate predictor of numerical features from categorical features, and vice versa. This approach does not make any assumption about the data distribution. To construct a predictor, many diﬀerent models such as Linear Regression [[7](#bookmark32)], SVM Regression [[11](#bookmark33)], and Neural Network [[32](#bookmark34)] can be used. Because of its robustness, we follow the second approach in this study.

Clustering Mixed-Type Data with Correlation-Preserving Embedding 345

**3 Related Work**

In literature, there are three main approaches for clustering mixed-type data.

The ﬁrst approach is ﬁnding a numerical representation of data then applying clustering algorithms designed for numerical data. Basic encoding techniques, i.e., onehot, ordinal, and binary encoding, are typically used to transform cat- egorical features into numerical features. The basic encoding approach is fast; however, it operates on individual features, hence, does not correctly diﬀerentiate between categorical values or capture the correlation between the features. Sev- eral techniques have been introduced to address these problems. Autoencoder [[3](#bookmark35)] takes the onehot encoded data as input to learn the compact representa- tion of data. However, autoencoder alone does not fully preserve the correla- tions between features. DEC [[41](#bookmark36)] is a variant of autoencoder, which simulta- neously learns feature representations and cluster assignments. DEC ﬁrst ini- tializes its parameters with a deep autoencoder, then optimizes them by iter- ating between computing an auxiliary target distribution and minimizing the Kullback–Leibler (KL) divergence [[19](#bookmark37)]. DEC focuses more on optimizing the discrimination between data objects. Similarly, MAI [[24](#bookmark38)] learns the pair-wise relationship between features and focuses on learning the discrimination between objects. It ﬁrst estimates the density between each pair of categorical and numer- ical feature. Then, each data object can be represented by a coupled encoding matrix. In addition, MAI also has another representation of data in the onehot encoding space. MAI takes these two representations as input and employs a neu- ral network to learn the data representation. MAI triggers the learning process by preserving the distance orders in every set of three data objects in the one- hot encoding and couple encoding space. This approach captures the pair-wise relationships between categorical and numerical features. However, it does not capture the relationship between more than two features. Besides, it preserves the order between data points in the onehot encoding space, which is generally not accurate. Moreover, the process of estimating the density of couplings is very time-consuming.

The second approach is converting numerical values into categorical values, then applying clustering techniques designed for categorical data. The numeri- cal features are discretized into equal-size bins. K-modes [[22](#bookmark14)], which is based on k-means, is a common clustering technique for categorical data. K-modes uses the Hamming distance, which is the number of features where two data objects diﬀer. K-modes tries to minimize the sum of intra-cluster Hamming distances from the mode of clusters to their members. The mode vector consists of cate- gorical values, each being the mode of an attribute. Several categorical clustering algorithms such as COOLCAT [[4](#bookmark12)] and LIMBO [[1](#bookmark11)] are based on minimizing the entropy of the whole arrangement. However, the numerical discretization process is information lossy, which can incur low clustering performance.

The third approach is applying algorithms designed for mixed-type data. ClicoT [[6](#bookmark16)] and Integrate [[9](#bookmark39)] cluster mixed-type data by minimizing the Huﬀ- man coding cost for coding numerical, categorical values, and model parame- ters. K-prototypes [[22](#bookmark14)] combines k-means for numerical features and k-modes

346 L. Tran et al.

[[22](#bookmark14)] for categorical features. It minimizes the intra-cluster distances, including the Euclidean distances for numerical features and the Hamming distances for categorical features. MDBSCAN in [[5](#bookmark40)] introduced distance hierarchy as a dis- tance measure suitable for categorical and numerical attributes and then applied a modiﬁed DBSCAN [[14](#bookmark8)] clustering.

**4 Correlation Preserving Embedding for Categorical Features - COPE**

Instead of clustering data X in the original space X, we propose to ﬁrst trans- form the onehot encoded values Bc ∈ Bc of categorical features Xc ∈ Xc with a non-linear mapping fθ : Bc → Zc where θ is a list of learnable parameters and Zc is the latent embedding space, then concatenate the numerical embedding with the normalized numerical features in Xn. In other words, the ﬁnal repre- sentation is [fθ (Bc ), Norm(Xn )], where Norm(.) is a normalization function to ensure numerical values to be in the same range with the categorical embedding. Such a representation allows to cluster the mixed-type input data with existing algorithms designed for clustering numerical data. The dimensions of Zc is typ- ically much smaller than Xc in order to avoid “the curse of dimensionality” [[23](#bookmark41)]. To parameterize fθ , Deep Neural Network (DNN) is a natural choice due to its theoretical function approximation property [[20](#bookmark42)] and its demonstrated feature learning capability [[8](#bookmark43)]. Following this approach, our proposed COPE network consists of two main components, i.e., a Deep Autoencoder [[29](#bookmark44)] and two Fully Connected Neural Networks (FCNN). Figure [1](#bookmark45) illustrates our COPE network design.

The Deep Autoencoder is a deep neural network eﬃcient in representation learning and is used to extract latent compact features from the categorical input data. It can capture the relationship between the categorical variables. It consists of two parts, the encoder and the decoder, which can be deﬁned as transitions Eφ and Dμ such that Eφ : Xc → Zc and Dμ : Zc → Xc. The learnable parameters φ, μ are optimized by minimizing the reconstruction loss:

Lae = ||Xc − (EoD)Xc ||2 (1)

We parameterize φ and μ by deep fully connected networks with l layers, respec- tively. In the encoder, after each layer, the number of units in a fully connected layer is decreased by α . In contrast, in the decoder, after each layer, the num- ber of units in a fully connected layer is increased by α . The embedding Zc is computed as follows:

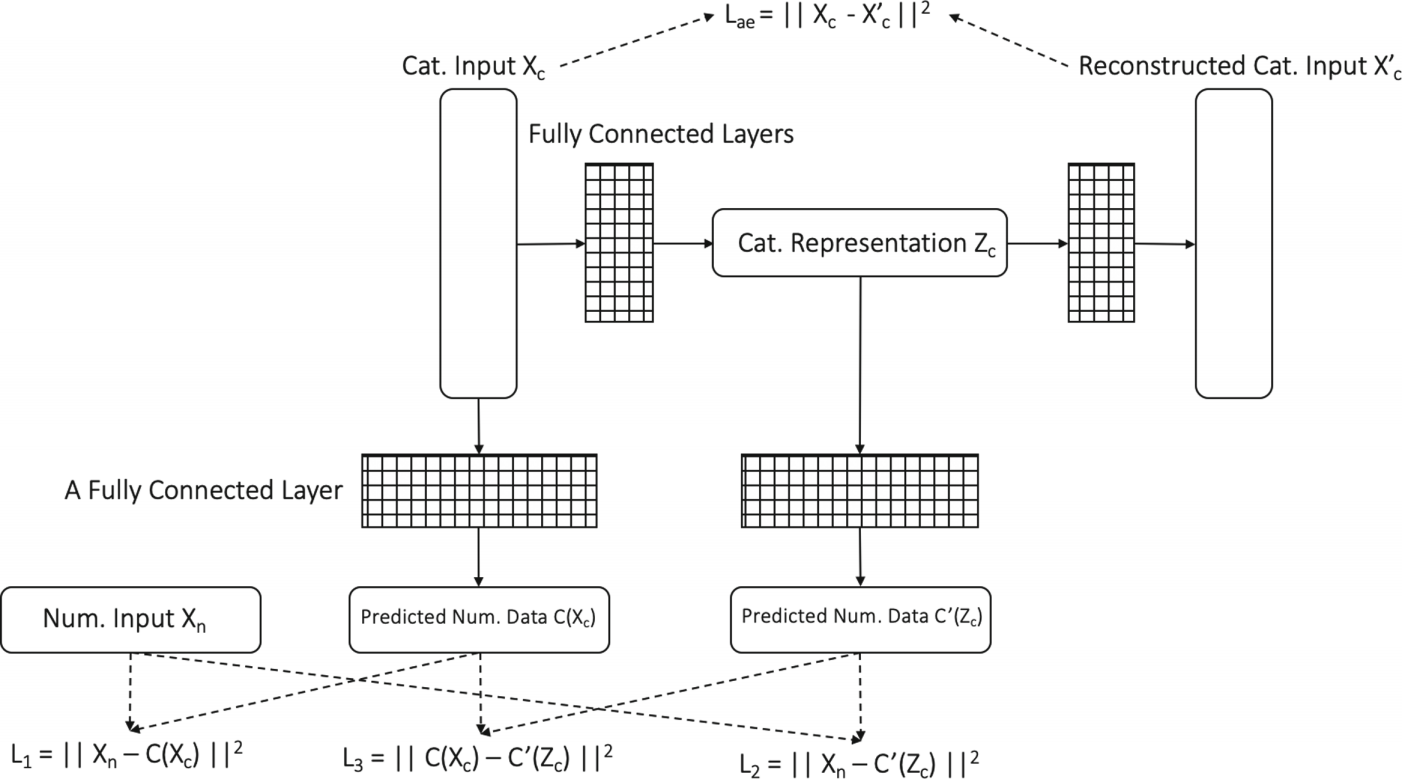
Zc = fl (fl−1(...(Xc ))) (2)

where

fi (x) = σ(Wi.x) (3) with σ is an activation function.

Our objective is to preserve the correlation between the two types of features from the plain encoding and the embedding space. Let Cψ : Xc → Xn and

Clustering Mixed-Type Data with Correlation-Preserving Embedding 347



**Fig. 1.** The Correlation-Preserving Representation Learning Network - COPE. It con- tains a Deep Autoencoder and two Fully Connected Neural Networks to learn the representation of categorical features.

Cω, : Zc → Xn be the functions mapping the categorical features in the plain

encoding space and embedding space to the numerical features, respectively. The parameters ψ, ω are optimized by minimizing the loss:

Lcr = L1 + L2 + L3 (4)

where

L1 = ||Xn −C (Xc )|| 2 , L2 = ||Xn −C, (Zc )|| 2 , L3 = ||C (Xc ) −C, (Zc )|| 2 (5)

The losses L1 and L2 constrain C and C, to learn the relationship between cate- gorical features in the plain encoding and embedding space with the numerical features, respectively. The loss L3 constrains Xc and Zc to have similar corre- lations with Xn. The intuition is that if L3 is small, C and C, will have similar performance in predicting Xn. Similar to the encoder and decoder, we param- eterize ψ and ω by one fully connected layer, respectively. To optimize all the parameters θ = {φ, μ, ψ, ω} concurrently, we combine the two loss Lae and Lcr into one target loss and use Adam optimizer [[25](#bookmark46)]:

L = Lae + Lcr (6)

After the parameters are optimized, the encoder is used to compute the embed- ding Zc from the categorical input data Xc. Then, the ﬁnal data representation [Zc, Norm(Xn )] is obtained. Consequently, clustering algorithms for numerical data such as k-means [[18](#bookmark9)] and DBSCAN [[14](#bookmark8)] can be applied on the ﬁnal data representation.

348 L. Tran et al.

**5 Experiment Results**

**5.1 Experimental Methodology**

We ﬁrst examined the correlation preserving and convergence capacity of COPE. Then we compared the performance of COPE with the baseline algorithms using the Adjusted Mutual Information [[36](#bookmark21)] and Fowlkes-Mallows [[15](#bookmark22)] scores. Finally, we compared the quality of the representations produced by algorithms using t-SNE visualization [[28](#bookmark23)].

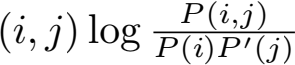
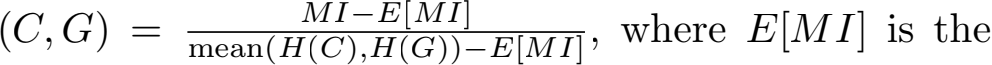
All the algorithms were implemented in Python. In COPE, the embedding dimension was set to a half of the onehot encoding dimensions for categorical features, and the parameter α was set to 1.2. In the Deep Autoencoder, in each fully connected layer, we apply asigmoid activation function. We used the Adam optimizer with a learning rate of 0.001.

**Baseline Algorithms.** The baseline algorithms were selected carefully for each mixed-type data clustering approach. For the ﬁrst approach of converting cate- gorical values into numerical values, we selected Onehot, Ordinal, Binary encod- ing as the standard encoding methods, Autoencoder (AE) as a typical represen- tation learning method, and MAI [[24](#bookmark38)] as a state-of-the-art method. For Autoen- coder, we used three variants, i.e., AE-cat, AE-all, and DEC [[41](#bookmark36)]. AE-cat takes only the one-hot encoded values of categorical values as input to learn the repre- sentation for categorical features. AE-all takes both numerical features and the one-hot encoded values of categorical values as input to learn the representation for mixed-type data directly. DEC [[41](#bookmark36)] is similar to AE-all but simultaneously learns feature representations and cluster assignments. We set the number of units in each layer of the encoder and decoder in these autoencoder variants similarly to COPE. For the second approach of converting numerical values into categorical values, we selected k-modes [[22](#bookmark14)] because of its popularity. Each numerical feature was discretized into ten equal-size bins. For the third app- roach of using a clustering algorithm designed for mixed-type data, we selected k-prototypes [[22](#bookmark14)] as a popular algorithm and ClicoT [[6](#bookmark16)] as the state-of-the-art algorithm.

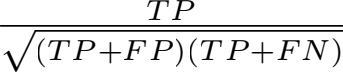
**Clustering Metrics.** In this study, we used the ground-truth classes to evaluate the performance of clustering algorithms. The assumption is that the members belong to the same classes are more similar than the members of diﬀerent classes. Hence, the goodness of a clustering assignment can be measured as its similarity to the ground-truth classes. We used the two widely used clustering metrics,i.e., Adjusted Mutual Information (AMI) [[36](#bookmark21)] and Fowlkes-Mallows Index (FMI) [[15](#bookmark22)].

**Adjusted Mutual Information.** Let C,G denote the cluster and ground- truth class assignments, respectively, of the same N data points. The entropy of C and G is deﬁned as follows: H (C) = −Σi| |1 P (i) log(P (i)), where P (i) = |Ci |/N, and H (G) = −Σi| |1 P, (i) log(P, (i)) where P, (i) = |Gi |/N.

Clustering Mixed-Type Data with Correlation-Preserving Embedding 349

The mutual information between C and G is calculated by: MI (C, G) = Σi| |1 Σj||1 P  . The value of mutual information is adjusted for chance as follows: AMI  expected value of the mutual information. The AMI value is ranged from 0 to 1. An AMI of 1 indicates two label assignments are equal.

**Fowlkes-Mallows Index.** The Fowlkes-Mallows index is deﬁned as the geo-

metric mean of pair wise precision and recall: FM I =  where

TP stands for True Positive, the number of pair of data points belonging to the same classes and clusters; FP stands for False Positive, the number of pair of data points belonging to the same classes but diﬀerent clusters; FN stands for False Negative, the number of pair of data points belonging to the same clusters but diﬀerent classes. The FMI value is ranged from 0 to 1. A FMI of 1 indicates

two label assignments are equal.

**5.2 Datasets**

We used six real-world UCI datasets [[13](#bookmark47)] in various domains. The statistics of the datasets are reported in Table [2](#bookmark48), including the dataset name, dataset size, the number of categorical features, numerical features, and classes. The KDD dataset was extracted from the original KDD Cup 99 dataset to obtain more balancing data. More speciﬁcally, we removed the classes with less than 1000 data points and randomly selected at most 10000 data points for each remaining class. The counts of classes areas follows: {‘back.’: 2203, ‘ipsweep.’: 1247, ‘neptune.’: 10000, ‘normal.’: 10000, ‘portsweep.’: 1040, ‘satan.’: 1589, ‘smurf.’: 10000, ‘warezclient.’: 1020} . For the Echo dataset, weremoved the class with only one instance. For all datasets, we imputed missing numerical values by mean values, and categorical values by a value denoted by word “missing” .

**Table 2.** Statistics of UCI datasets

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Datasets | Description | Size | dc | dn | Class |
| KDD | Network packages of diﬀerent attacks | 37099 | 7 | 34 | 8 |
| Income | Census income data | 32561 | 9 | 6 | 2 |
| ACA | Australian credit approval data | 690 | 8 | 6 | 2 |
| CRX | Credit card applications | 690 | 9 | 6 | 2 |
| Titanic | Titanic’s passenger information | 891 | 5 | 4 | 2 |
| Echo | Patients with heart attack | 131 | 2 | 8 | 2 |

**5.3 COPE - Correlation Preservation and Convergence Test**

We ﬁrst examined the correlation preserving capacity and the convergence of our proposed method, COPE.

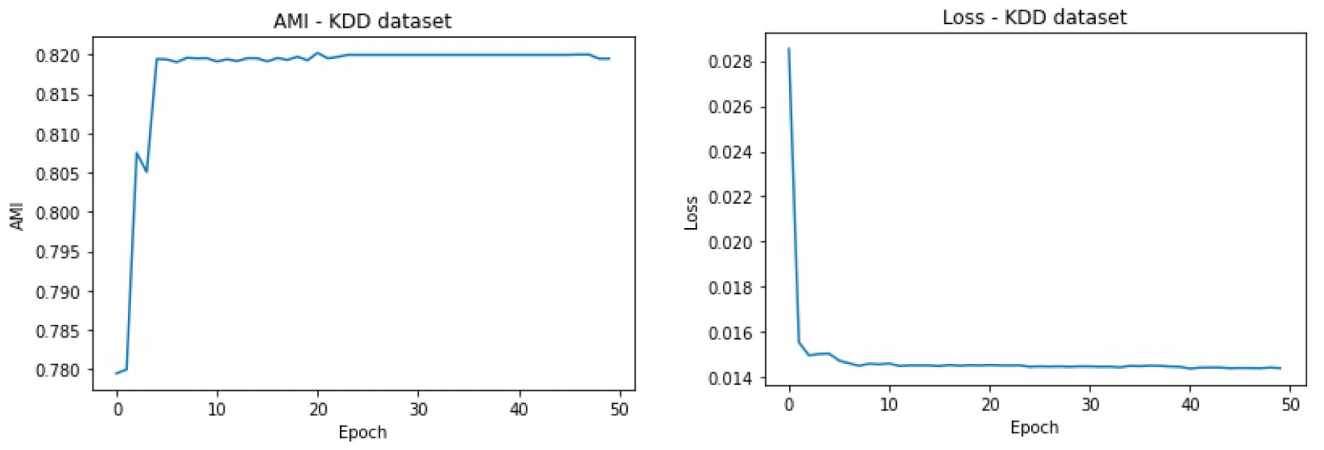
350 L. Tran et al.

**Correlation Preservation.** Table [3](#bookmark49) reports the training losses of COPE for all the datasets. As can be seen in this table, the losses L1 and L2 are small. It shows that the mapping functions C and C, well represents the relationship between categorical and numerical attributes in both the plain and embedding space. The loss L3 is very small for all the datasets, which proves the correlation preserving capacity of COPE. The autoencoder loss Lae is small, which shows that the original data can be well reconstructed from the embedded data.

**Table 3.** Training losses for all datasets.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Datasets | L1 | L2 | L3 | Lae |
| KDD | 0.0058 | 0.0058 | 2.29E−07 | 3.35E−05 |
| Titanic | 0.0024 | 0.0017 | 1.17E−05 | 1.30E−04 |
| CRX | 0.012 | 0.0118 | 6.67E−06 | 6.50E−04 |
| Income | 0.0083 | 0.0083 | 2.42E−07 | 1.07E−04 |
| ACA | 0.009 | 0.008 | 3.02E−06 | 5.30E−05 |
| Echo | 0.0296 | 0.0294 | 1.54E−08 | 1.40E−04 |

**Convergence Test.** We applied k-means clustering [[18](#bookmark9)] on the learned repre- sentation and reported how the loss and AMI change across epochs. We used the number of ground-truth classes to set the parameter k - the number of clusters. Figure [2](#bookmark50) reports the AMI and the training loss with the KDD dataset in 50 epochs. Similar results can be obtained with the other datasets. As can be seen in this ﬁgure, the AMI and total training loss converged within 10 epochs.



(a) AMI (b) Training Loss **Fig. 2.** Convergence test on dataset KDD.

Clustering Mixed-Type Data with Correlation-Preserving Embedding 351

**5.4 Clustering Results**

We applied k-means clustering [[18](#bookmark9)] on the data representation produced by the methods producing numerical representation and compared their AMIs and FMIs. We also compared them with ClicoT [[6](#bookmark16)] and k-prototypes [[22](#bookmark14)] designed for mixed-type data, and k-modes [[22](#bookmark14)] desgined for categorical data. The number of clusters k in k-means, k-modes, and k-prototypes, was set to be the num- ber of ground-truth classes. The results are reported in Tables [4](#bookmark51) and [5](#bookmark52) with all the datasets. As reported in these tables, the AMI and FMI of COPE were the highest for all the datasets. The top three performers were COPE, MAI, and DEC, which produce numerical representation. On average, COPE demon- strated approximately 37% and 30% improvement in AMI over DEC and MAI, respectively. MAI and DEC optimized the discrimination between data points and did not well preserve the relationship between categorical and numerical features. In most cases, we observed AE-all outperforms AE-cat because AE-all is able to integrate all features. However, for the KDD and Income datasets, we observed AE-cat performs better than AE-all. This is because the correla- tion between categorical and numerical features was not capture correctly in AE-all. ClicoT automatically determines the number of clusters, which might be diﬀerent from the number of ground-truth classes. K-prototypes uses Ham- ming distance for categorical features, which does not capture the relationship between categorical and numerical features and oﬀers the lowest performance. The basic encoding methods, i.e., Onehot, Ordinal, and Binary, do not con- sider any relationship between features, hence, they also oﬀered low clustering performances.

**Table 4.** Clustering performance - AMI. The results of COPE and numerical repre- sentation methods are obtained using k-means.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datasets | Numerical representa- tion | | | | | | | Categorical representa- tion | Designed for  mixed-type data | | COPE |
| Onehot | Ordinal | Binary | MAI | AE-all | AE-cat | DEC | K-modes | ClicoT | K-prototypes |
| KDD | 0.77 | 0.64 | 0.79 | 0.71 | 0.76 | 0.77 | 0.73 | 0.71 | 0.74 | 0.72 | **0.82** |
| Income | 0.11 | 0.02 | 0.10 | 0.13 | 0.11 | 0.13 | 0.13 | 0.09 | 0.03 | 0.00 | **0.17** |
| ACA | 0.43 | 0.02 | 0.01 | 0.43 | 0.36 | 0.22 | 0.16 | 0.23 | 0.18 | 0.28 | **0.44** |
| CRX | 0.02 | 0.02 | 0.02 | 0.43 | 0.16 | 0.01 | 0.43 | 0.20 | 0.18 | 0.03 | **0.44** |
| Titanic | **0.23** | 0.02 | **0.23** | 0.06 | **0.23** | 0.08 | **0.23** | 0.09 | 0.07 | 0.08 | **0.23** |
| Echo | 0.11 | 0.21 | 0.01 | 0.32 | 0.44 | 0.01 | 0.44 | 0.11 | 0.26 | 0.37 | **0.59** |
| Average | 0.28 | 0.16 | 0.19 | 0.37 | 0.34 | 0.20 | 0.35 | 0.24 | 0.24 | 0.15 | **0.48** |

**5.5 Data Representation Analysis**

There are several approaches for data visualization such as PCA [[39](#bookmark53)], and t-SNE [[28](#bookmark23)]. They are both used for dimensionality reduction. While PCA is a linear projection, t-SNE uses the local relationship between data points to create a

352 L. Tran et al.

**Table 5.** Clustering performance - FMI. The results of COPE and numerical repre- sentation methods are obtained using k-means.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Datasets | Numerical representa- tion | | | | | | | Categorical representa- tion | Designed for  mixed-type data | | COPE |
| Onehot | Ordinal | Binary | MAI | AE-all | AE-cat | DEC | K-modes | ClicoT | K-prototypes |
| KDD | 0.78 | 0.68 | 0.82 | 0.72 | 0.77 | 0.78 | 0.75 | 0.73 | 0.77 | 0.77 | **0.83** |
| Income | 0.64 | 0.32 | 0.34 | 0.65 | 0.56 | 0.65 | 0.65 | 0.34 | 0.24 | 0.41 | **0.67** |
| ACA | 0.75 | 0.61 | 0.53 | 0.75 | 0.71 | 0.65 | 0.61 | 0.68 | 0.41 | 0.69 | **0.77** |
| CRX | 0.56 | 0.52 | 0.56 | 0.54 | 0.62 | 0.50 | 0.75 | 0.64 | 0.47 | 0.68 | **0.77** |
| Titanic | **0.69** | 0.52 | **0.69** | 0.54 | **0.69** | 0.63 | **0.69** | 0.61 | 0.38 | 0.08 | **0.69** |
| Echo | 0.56 | 0.52 | 0.53 | 0.79 | 0.80 | 0.53 | 0.80 | 0.64 | 0.71 | 0.37 | **0.88** |
| Average | 0.66 | 0.53 | 0.58 | 0.67 | 0.69 | 0.62 | 0.71 | 0.61 | 0.50 | 0.50 | **0.77** |



|  |
| --- |
|  |

(a) COPE (b) AE-all (c) MAI

|  |
| --- |
|  |

(d) Onehot (e) AE-cat (f) DEC

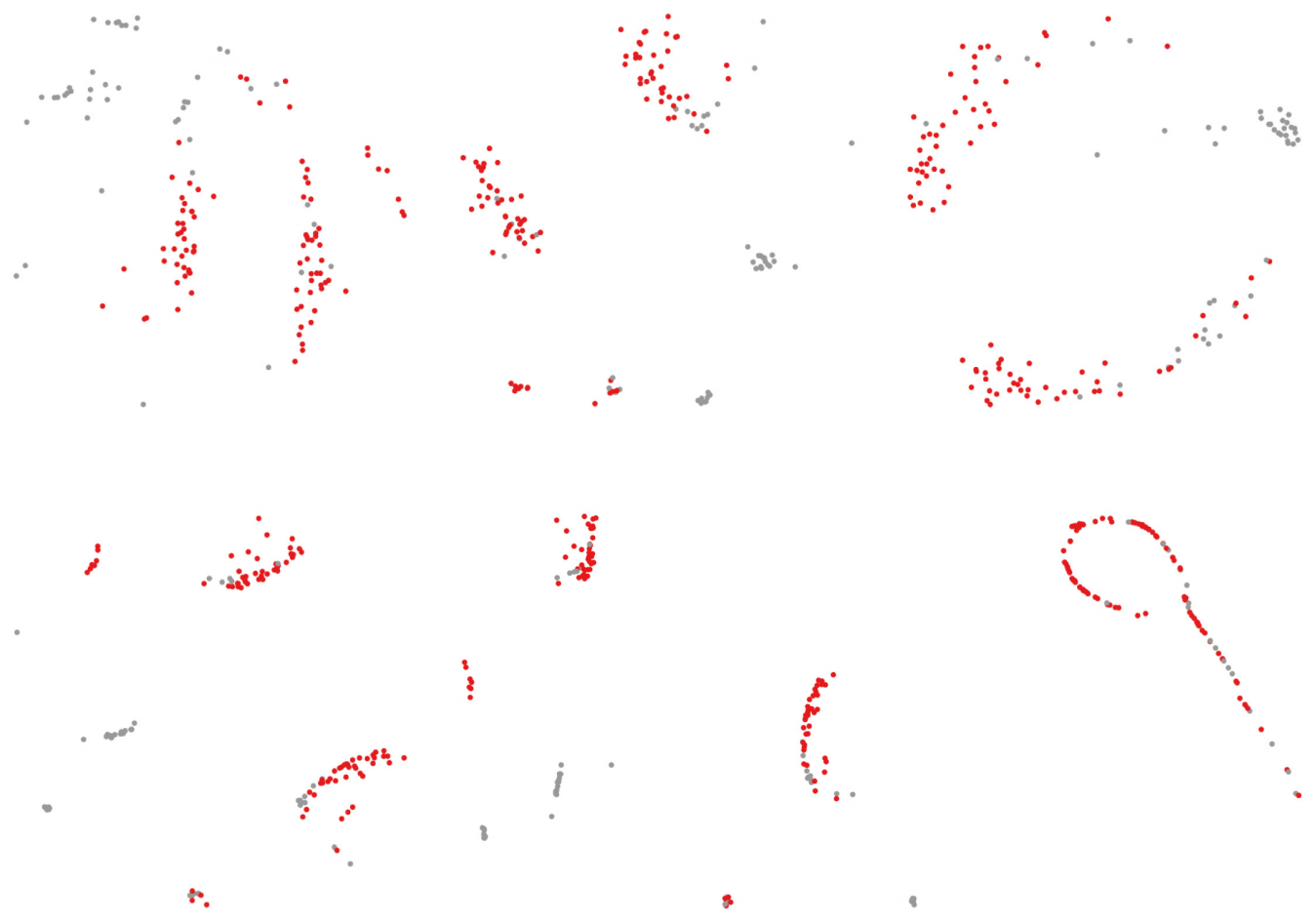
**Fig. 3.** The t-SNE visualization of data representations on the KDD dataset. (Color ﬁgure online)

low-dimensional mapping. Similar data points in the original space tend to have small distances in the t-SNE mapping. Because of the capacity to capture non- linear dependencies, we adopted t-SNE to compare the quality of the mixed-type data representations.

Here, we compared the top six methods that provide numerical representa- tions, i.e., COPE, DEC, MAI, AE-all, Onehot, and AE-cat. In t-SNE, we set the perplexity to be 100, and the number of iterations to be 5000. Figures [3](#bookmark54) and [4](#bookmark55) illustrate the representations of the methods for the KDD and Echo datasets,

Clustering Mixed-Type Data with Correlation-Preserving Embedding 353

|  |
| --- |
|  |

(a) COPE (b) AE-all (c) MAI

|  |
| --- |
|  |

(d) Onehot (e) AE-cat (f) DEC

**Fig. 4.** The t-SNE visualization of data representations on the Echo dataset. (Color ﬁgure online)

respectively, in a two dimensional t-SNE mapping. The representations of the other datasets are in the Appendix section. Diﬀerent colors represent diﬀerent ground-truth classes.

The KDD dataset with eight diﬀerent classes is plotted in red, blue, green, grey, yellow, pink, purple, and brown colors. As shown in Figure [3](#bookmark54), in COPE, most data points with the same classes are grouped into clusters, and the clusters are separated quite clearly. The three largest groups are green, pink, and purple. Meanwhile, in the other methods, we only observed at most ﬁve major classes clearly with green, purple, yellow, brown, and red colors. Many data points in other colors, e.g., pink data points, were hidden among green and purple data points. Note that the group of pink data points is the largest group in the KDD dataset. It shows the much better separability of COPE compared to other methods. In MAI and DEC, we observed quite clearly groups of data points. However, each group consists of data points indiﬀerent classes. In Onehot and AE-cat, more data points were scattered because they do not consider the relationship between categorical and numerical features.

For the Echo dataset, which has two diﬀerent classes, as shown in Figure [4,](#bookmark55) there are two big groups of red data points in all methods. Similar to the KDD dataset, COPE has the fewest number of grey data points, which were falsely grouped with the red data points. The second best representation is AE-all. That explains why COPE and AE-all are the top two performers for the Echo dataset in clustering.

354 L. Tran et al.

**6 Conclusions**

In this paper, we proposed COPE,a framework to learn representation for mixed-

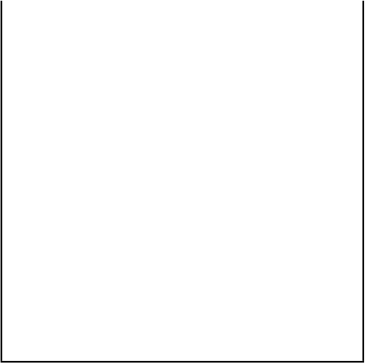
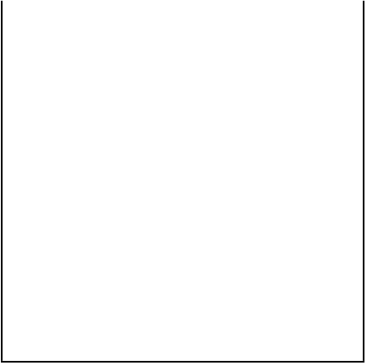
type data. It learns the embedding for categorical features using an autoencoder

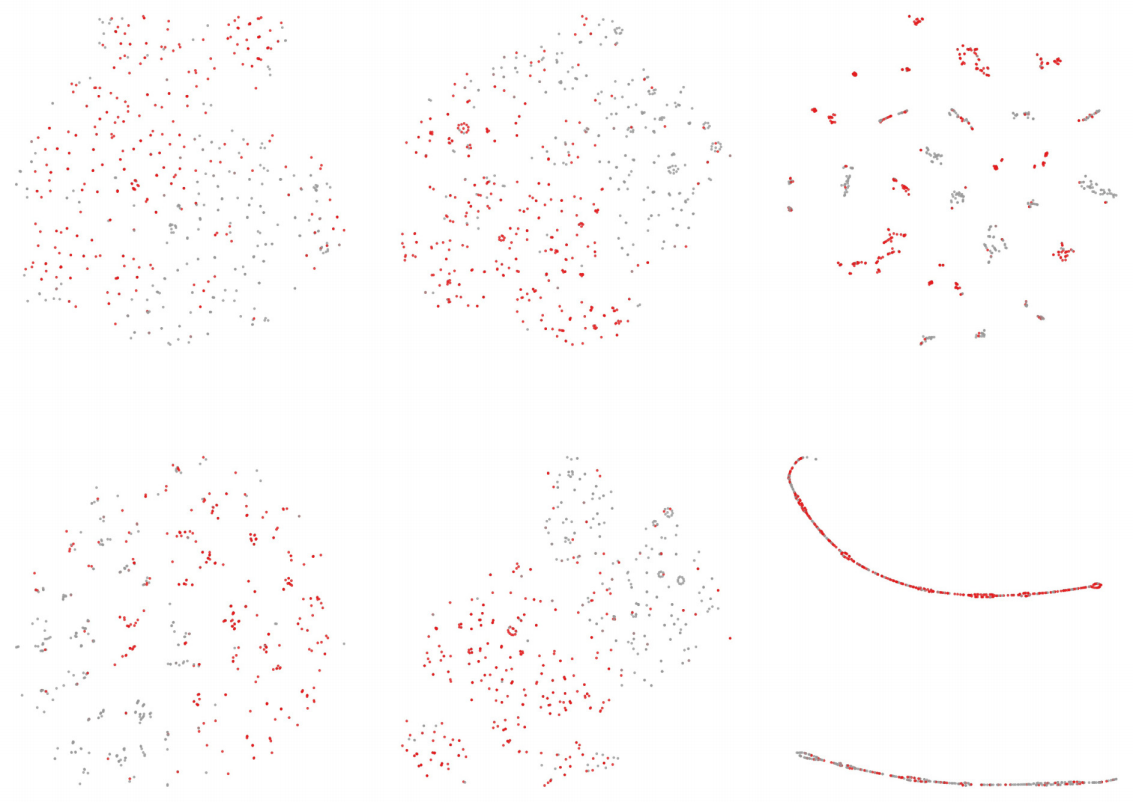
and two sub-networks to preserve the correlation between categorical and numer- ical features. We showed that COPE generates higher quality representation and oﬀers better clustering results than the competing methods by more than 30% in widely used clustering metrics. As future work, COPE can be combined with techniques that reﬁne the representation to enhance the discrimination between objects as in DEC for further improvement of the representation quality. We can also learn the embedding for numerical features using COPE by switching the roles of numerical and categorical features.

**Acknowledgement.** This work has been supported in part by NSF CNS-1951430, the USC Integrated Media Systems Center, and unrestricted cash gifts from Microsoft and Google. The opinions, ﬁndings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reﬂect the views of the sponsors.

**Appendix**

See Figs. [5,](#bookmark56) [6,](#bookmark57) [7](#bookmark58) and [8](#bookmark59)



(a) COPE (b) AE-all (c) MAI

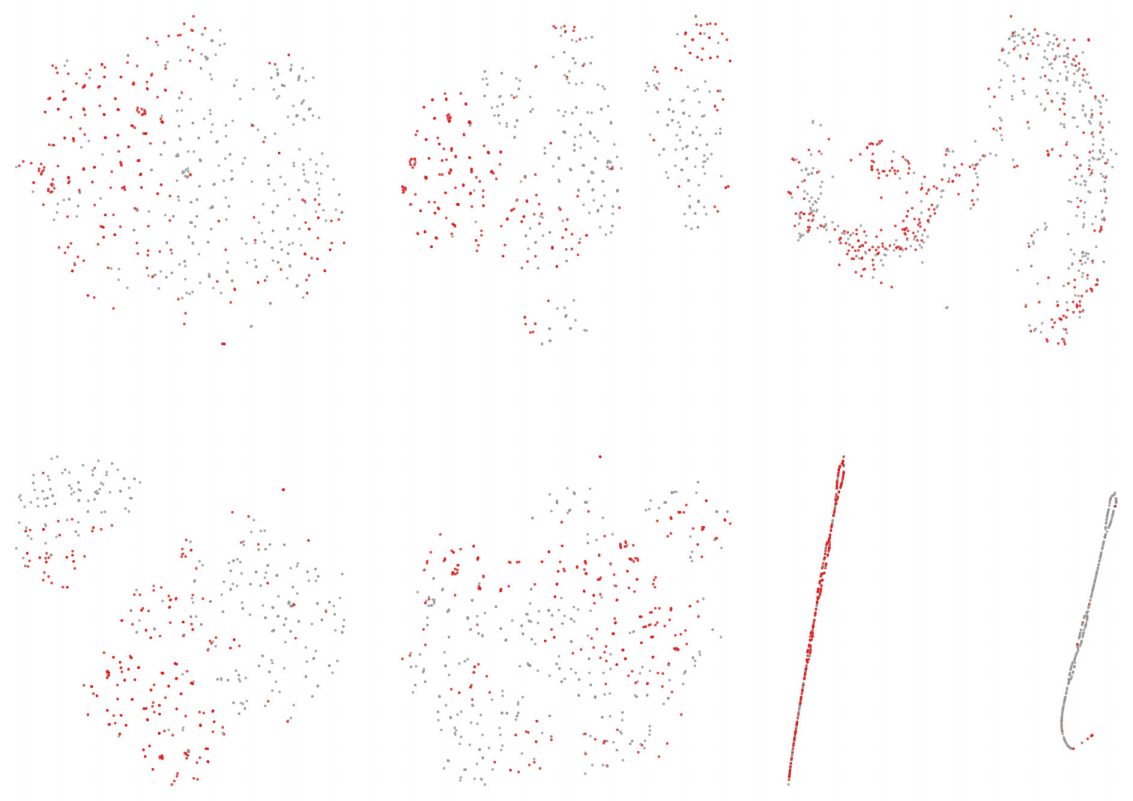
|  |
| --- |
|  |

(d) Onehot (e) AE-cat (f) DEC

**Fig. 5.** The t-SNE visualization of data representations on the ACA dataset.

Clustering Mixed-Type Data with Correlation-Preserving Embedding 355

|  |
| --- |
|  |

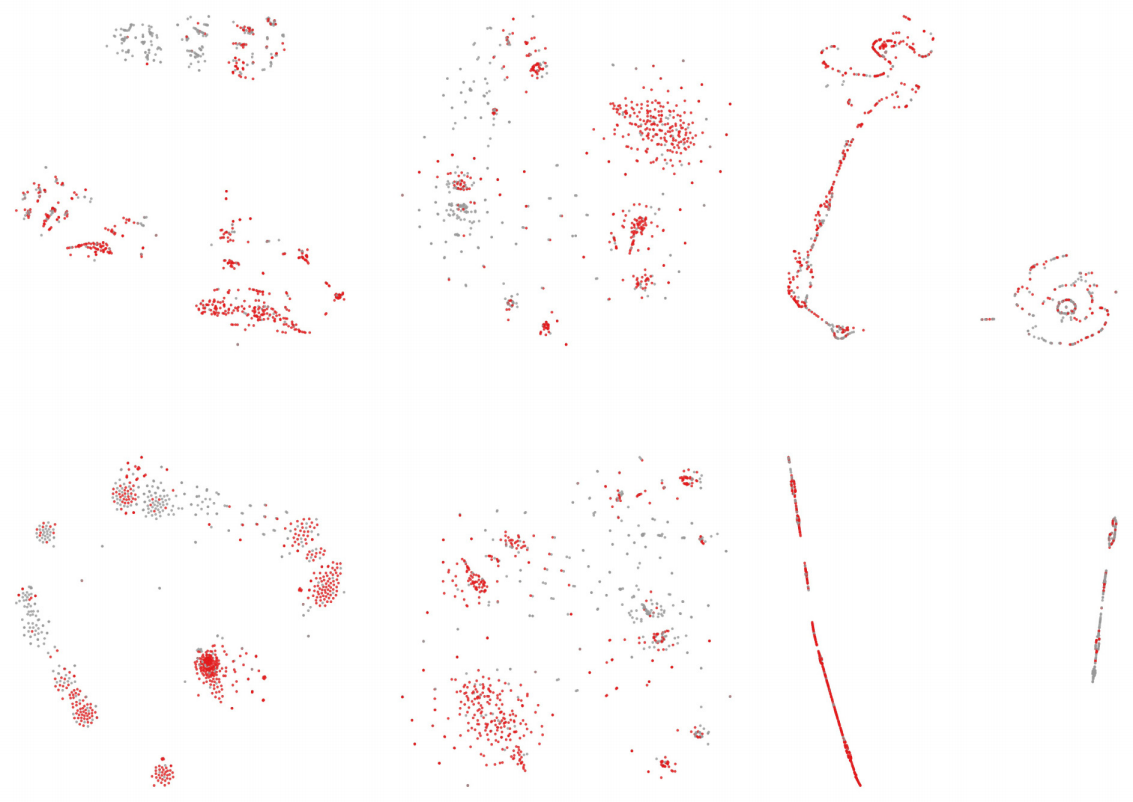
(a) COPE (b) AE-all (c) MAI

|  |
| --- |
|  |

(d) Onehot (e) AE-cat (f) DEC

**Fig. 6.** The t-SNE visualization of data representations on the CRX dataset.

|  |
| --- |
|  |

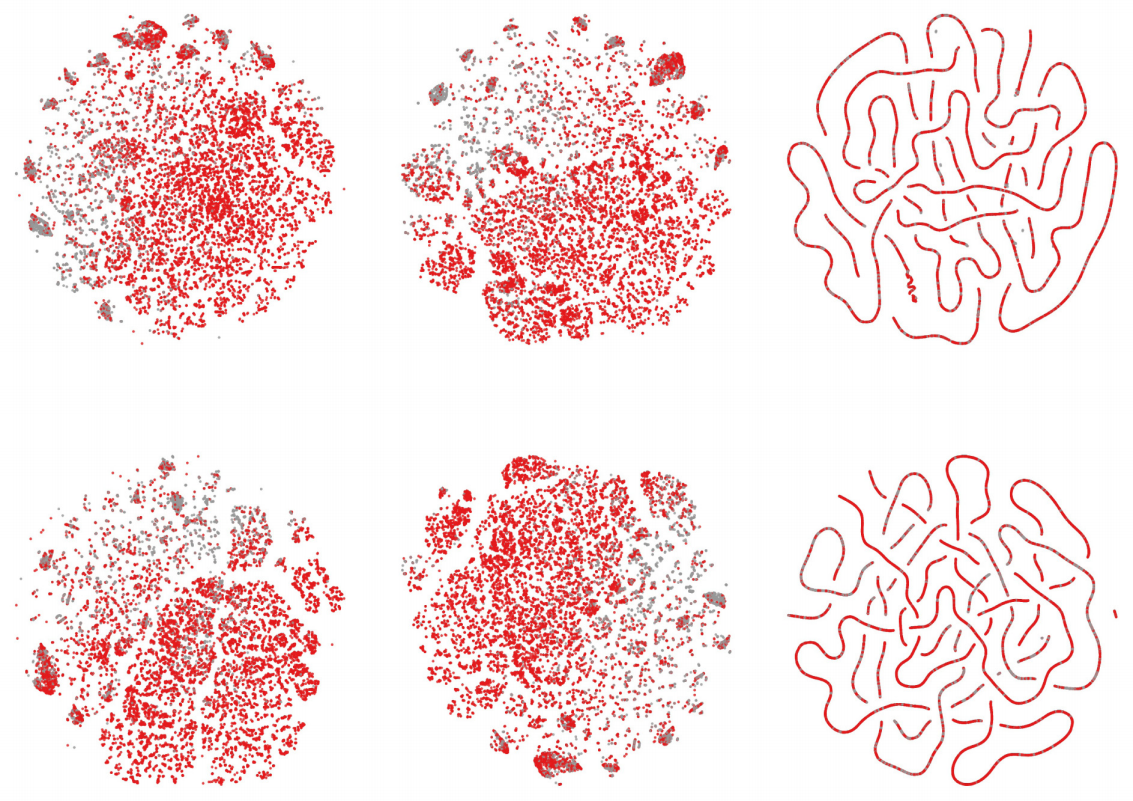
(a) COPE (b) AE-all (c) MAI

|  |
| --- |
|  |

(d) Onehot (e) AE-cat (f) DEC

**Fig. 7.** The t-SNE visualization of data representations on the Titanic dataset.

356 L. Tran et al.



|  |
| --- |
|  |

(a) COPE (b) AE-all (c) MAI

|  |
| --- |
|  |

(d) Onehot (e) AE-cat (f) DEC

**Fig. 8.** The t-SNE visualization of data representations on the Income dataset.

**References**

1. Andritsos, P., Tsaparas, P., Miller, R.J., Sevcik, K.C.: LIMBO: scalable clustering of categorical data. In: Bertino, E., et al. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 123–146. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-24741-](https://doi.org/10.1007/978-3-540-24741-8_9) [8 9](https://doi.org/10.1007/978-3-540-24741-8_9)

2. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: ordering points to identify the clustering structure. ACM SIGMOD Rec. **28**(2), 49–60 (1999)

3. Aytekin, C., Ni, X., Cricri, F., Aksu, E.: Clustering and unsupervised anomaly detection with l 2 normalized deep auto-encoder representations. In: 2018 Interna- tional Joint Conference on Neural Networks (IJCNN), pp. 1–6. IEEE (2018)

4. Barbar, D., Li, Y., Couto, J.: Coolcat: an entropy-based algorithm for categorical clustering. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 582–589 (2002)

5. Behzadi, S., Ibrahim, M.A., Plant, C.: Parameter free mixed-type density-based clustering. In: Hartmann, S., Ma, H., Hameurlain, A., Pernul, G., Wagner, R.R. (eds.) DEXA 2018. LNCS, vol. 11030, pp. 19–34. Springer, Cham (2018). [https://](https://doi.org/10.1007/978-3-319-98812-2_2) [doi.org/10.1007/978-3-319-98812-2 2](https://doi.org/10.1007/978-3-319-98812-2_2)

6. Behzadi, S., Mller, N.S., Plant, C., Bhm, C.: Clustering of mixed-type data con- sidering concept hierarchies. In: Paciﬁc-Asia Conference on Knowledge Discovery and Data Mining, pp. 555–573. Springer (2019)

7. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coeﬃcient. In: Noise Reduction in Speech Processing. Springer Topics in Signal Processing, vol. 2,pp. 1–4. Springer, Heidelberg (2009). [https://doi.org/10.1007/978-3-642-00296-](https://doi.org/10.1007/978-3-642-00296-0_5) [0 5](https://doi.org/10.1007/978-3-642-00296-0_5)

8. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)

Clustering Mixed-Type Data with Correlation-Preserving Embedding 357

9. Bhm, C., Goebl, S., Oswald, A., Plant, C., Plavinski, M., Wackersreuther, B.:

Integrative parameter-free clustering of data with mixed type attributes. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS (LNAI), vol. 6118, pp. 38–47. Springer, Heidelberg (2010). [https://doi.org/10.1007/978-3-642-](https://doi.org/10.1007/978-3-642-13657-3_7) [13657-3 7](https://doi.org/10.1007/978-3-642-13657-3_7)

10. Cao, F., et al.: An algorithm for clustering categorical data with set-valued features. IEEE Trans. Neural Networks Learning Syst. **29**(10), 4593–4606 (2017)

11. Cherkassky, V., Ma, Y.: Practical selection of SVM parameters and noise estima- tion for SVM regression. Neural Netw. **17**(1), 113–126 (2004)

12. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space anal- ysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)

13. Dua, D., Graﬀ, C.: UCI machine learning repository (2017). [http://archive.ics.uci.](http://archive.ics.uci.edu/ml) [edu/ml](http://archive.ics.uci.edu/ml)

14. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. KDD **96**, 226–231 (1996)

15. Fowlkes, E.B., Mallows, C.L.: A method for comparing two hierarchical clusterings. J. Am. Stat. Assoc. **78**(383), 553–569 (1983)

16. Guha, S., Rastogi, R., Shim, K.: Rock: a robust clustering algorithm for categorical attributes. Inf. Syst. **25**(5), 345–366 (2000)

17. Hamka, F., Bouwman, H., De Reuver, M., Kroesen, M.: Mobile customer seg- mentation based on smartphone measurement. Telematics Inform. **31**(2), 220–227 (2014)

18. Hartigan, J.A., Wong, M.A.: Algorithm as 136: a k-means clustering algorithm. J. Royal Stat. Soci. Series c (applied statistics) **28**(1), 100–108 (1979)

19. Hershey, J.R., Olsen, P.A.: Approximating the kullback leibler divergence between Gaussian mixture models. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07, vol. 4, pp. IV-317. IEEE (2007)

20. Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Netw. **4**(2), 251–257 (1991)

21. Hosmer Jr., D.W., Lemeshow, S., Sturdivant, R.X.: Applied logistic regression, vol.

398. Wiley (2013)

22. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Disc. **2**(3), 283–304 (1998)

23. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of The Thirtieth Annual ACM Symposium on Theory of Computing, pp. 604–613. ACM (1998)

24. Jian, S., Hu, L., Cao, L., Lu, K.: Metric-based auto-instructor for learning mixed data representation. In: Thirty-Second AAAI Conference on Artiﬁcial Intelligence (2018)

25. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](http://arxiv.org/abs/1412.6980) (2014)

26. Kohavi, R.: Scaling up the accuracy of Naive-bayes classiﬁers: a decision-tree hybrid. KDD **96**, 202–207 (1996)

27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classiﬁcation with deep con- volutional neural networks. In: Advances in Neural Information Processing Sys- tems, pp. 1097–1105 (2012)

28. van der Maaten, L., Hinton, G.: Visualizing data using t-sne. J. Mach. Learn. Res. **9**, 2579–2605 (2008)

358 L. Tran et al.

29. Marchi, E., Vesperini, F., Eyben, F., Squartini, S., Schuller, B.: A novel app- roach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1996–2000. IEEE (2015)

30. Ni, X., Quadrianto, N., Wang, Y., Chen, C.: Composing tree graphical models with persistent homology features for clustering mixed-type data. In: International Conference on Machine Learning, pp. 2622–2631 (2017)

31. Salem, S.B., Naouali, S., Chtourou, Z.: A fast and eﬀective partitional clustering algorithm for large categorical datasets using a k-means based approach. Comput. Electr. Eng. **68**, 463–483 (2018)

32. Specht, D.F., et al.: A general regression neural network. IEEE Trans. Neural Networks **2**(6), 568–576 (1991)

33. Tate, R.F.: Correlation between a discrete and a continuous variable. point-biserial correlation. Ann. Math. Stat. **25**(3), 603–607 (1954)

34. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD cup 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. IEEE (2009)

35. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th Interna- tional Conference on Machine Learning, pp. 1096–1103 (2008)

36. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**, 2837–2854 (2010)

37. Von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)

38. Wiwie, C., Baumbach, J., Rttger, R.: Comparing the performance of biomedical clustering methods. Nat. Methods **12**(11), 1033 (2015)

39. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemom. Intell. Lab. Syst. **2**(1–3), 37–52 (1987)

40. Wu, Y., Duan, H., Du, S.: Multiple fuzzy c-means clustering algorithm in medical diagnosis. Technol. Health Care **23**(s2), S519–S527 (2015)

41. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp. 478–487 (2016)