Embedding-based Representation of Categorical Data by Hierarchical Value Coupling Learning

Songlei Jian[*][†][‡], Longbing Cao[*], Guansong Pang[*], Kai Lu[†][‡], Hang Gao[†]
[*]Advanced Analytics Institute, University of Technology Sydney, Australia
[†]Science and Technology on Parallel and Distributed Processing Laboratory,
[‡]State Key Laboratory of High Performance Computing,
College of Computer, National University of Defense Technology, China
{songlei.jian;longbing.cao}@uts.edu.au, guansong.pang@student.uts.edu.au

Abstract

Learning the representation of categorical data with hierarchical value coupling relationships is very challenging but critical for the effective analysis and learning of such data. This paper proposes a novel coupled unsupervised categorical data representation (CURE) framework and its instantiation, i.e., a coupled data embedding (CDE) method, for representing categorical data by hierarchical value-to-value cluster coupling learning. Unlike existing embedding- and similarity-based representation methods which can capture only a part or none of these complex couplings, CDE explicitly incorporates the hierarchical couplings into its embedding representation. CDE first learns two complementary feature value couplings which are then used to cluster values with different granularities. It further models the couplings in value clusters within the same granularity and with different granularities to embed feature values into a new numerical space with independent dimensions. Substantial experiments show that CDE significantly outperforms three popular unsupervised embedding methods and three state-of-the-art similarity-based representation methods.

1 Introduction

Categorical data with finite unordered feature values is ubiquitous in real-world applications and has received increasing attention for representation and learning [Wang *et al.*, 2015; Zhang *et al.*, 2015]. Unlike numerical data, categorical data cannot be directly manipulated per algebraic operations; hence many popular numerical learning algorithms are not directly applicable. Accordingly, it is important to learn an expressive numerical representation of categorical data.

In general, a good representation should effectively capture the intrinsic data characteristics [Bengio *et al.*, 2013]. One key characteristic in complex categorical data is the following hierarchical couplings (i.e., dependency or correlation) embedded in feature values. (1) On the low level, there exist strong couplings [Cao, 2015] between feature values, demonstrating the natural clusters of values. Taking census data as an example, it may be visible that the value *PhD* of feature *Education* is highly coupled with the values *Scientist* and *Professor* of feature *Occupation*; and these values form a semantic value cluster that characterizes one type of strong relation between education and occupation. In addition, different value clusters exist on different granularities and with different semantics [Foss and Zaïane, 2002]; e.g., all values belong to one super cluster at the coarsest granularity while each value is a cluster at the finest granularity. (2) On the high level, the clusters of feature values are further coupled with each other. Couplings exist between clusters of the same granularity and between clusters of different granularities.

For the above hierarchical value couplings in categorical data, existing embedding and similarity-based representation methods can capture only a part or none of these feature value couplings. Typical embedding-based representation methods transform categorical data to numerical data by encoding schemes, e.g., 0-1 encoding and Inverse Document Frequency (IDF) encoding [Aizawa, 2003; Pang *et al.*, 2016c]. These methods are easy to implement, but do not consider the couplings between feature values since they usually treat features independently. Some recent similarity-based representation methods, e.g., in [Ahmad and Dey, 2007; Ienco *et al.*, 2012; Wang *et al.*, 2015; Jia *et al.*, 2016] incorporate feature relations into similarity or kernel matrices. However, they do not capture the value clusters or the couplings between value clusters, leading to insufficient representation power in handling data with such hierarchical value couplings.

The hierarchical value couplings reflect the intrinsic data characteristics and complexities, which need to be captured in data representation. However, this is not a trivial task and, to our best knowledge, no work reported properly handles them. Accordingly, this paper aims to explicitly learn these couplings in terms of a new embedding-based representation. The main idea and contributions are as follows.

- A Coupled Unsupervised categorical data REpresentation (CURE) framework is proposed, which has a hierarchical learning structure. CURE is data-driven, which first learns the value clusters with different granularities by involving different low-level feature value couplings. It further generates an object embedding based on the concatenation of value embedding obtained by modeling couplings among the learned value clusters. In this way, CURE captures the intrinsic data characteristics and enables an effective numerical representation for categorical data with sophisticated couplings.

- The CURE framework is further instantiated into a Coupled Data Embedding (CDE) method to capture two types of value couplings. CDE captures complementary feature value couplings and produces diverse sets of informative value clusters. It further models the affluent couplings among these value clusters to embed categorical data into a new space with independent dimensions and rich semantics. As a result, CDE enables algebraic operations of categorical data in the Euclidean space.

Substantial experiments show that (1) CDE significantly outperforms three popular embedding methods and three state-of-the-art coupled similarity measures in terms of F-score for clustering on 10 real-world data sets with different value coupling complexities; (2) CDE performs stably and is insensitive to its parameters.

2 Related Work

This section discusses three closely related work, including embedding-based representation, similarity-based representation and coupling learning.

Embedding-based Representation. Embedding-based representation constructs a numerical vector for each categorical object. Encoding methods are commonly used for categorical data representation [Cohen *et al.*, 2013]. One popular method is the 0-1 encoding which encodes each feature value with a 0-1 indicator vector [Pang *et al.*, 2016c]. Although 0-1 coding is reversible with the original data, it assumes that the distance among all values equal 1 which is often violated in real-world data. To alleviate the curse of dimensionality issue, dimension reduction methods, like principal component analysis (PCA) [Jolliffe, 2002], are often conducted on a 0-1 encoding matrix. Another well-known method is IDF encoding [Aizawa, 2003] which differentiates the values from the same feature according to value frequency; however, it cannot capture the couplings between values from different features.

Several effective embedding methods are available for textual data, such as latent semantic indexing (LSI) [Deerwester *et al.*, 1990], latent Dirichlet allocation (LDA) [Blei *et al.*, 2003], skip-gram [Mikolov *et al.*, 2013a] and their variants [Hofmann, 1999; Wilson and Chew, 2010; Mikolov *et al.*, 2013b]. However, categorical data has an explicit feature structure, which is very different from unstructured textual data. Hence, these methods do not fit our target problem.

Similarity-based Representation. Similarity-based representation approaches (including some kernel methods) represent categorical data with an object similarity matrix. Various similarity measures have been designed to capture value couplings in data: ALGO [Ahmad and Dey, 2007] first use conditional probability of two feature values to describe the value couplings; DILCA [Ienco *et al.*, 2012] and DM [Jia *et al.*, 2016] incorporate feature selection and feature weighting into capturing feature couplings respectively; COS [Wang *et al.*, 2015] takes inter- and intra-feature couplings into consideration. Although these similarity measures capture the pairwise value couplings, they do not consider the value clusters and the couplings between value clusters. Meanwhile, similarity measurement is not an efficient method of representation since it must calculate and store an object similarity matrix which may limit its applications.

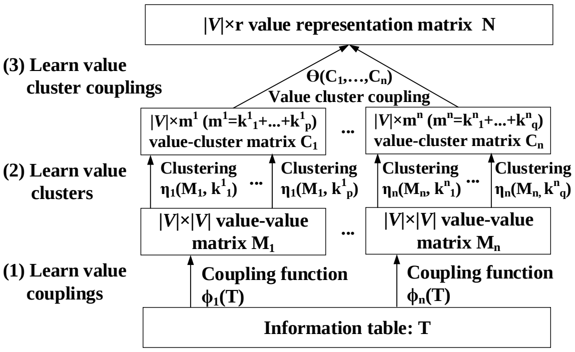
|V|×r value representation matrix N
(3) Learn value cluster couplings
Θ(C1,…,Cn)
Value cluster coupling
|V|×m^1 (m^1=k^1_1+...+k^1_p) value-cluster matrix C1
...
|V|×m^n (m^n=k^n_1+...+k^n_q) value-cluster matrix Cn
(2) Learn value clusters
Clustering η1(M1, k^1_1)
...
Clustering η1(M1, k^1_p)
Clustering ηn(Mn, k^n_1)
...
Clustering ηn(Mn, k^n_q)
|V|×|V| value-value matrix M1
...
|V|×|V| value-value matrix Mn
(1) Learn value couplings
Coupling function ϕ1(T)
Coupling function ϕn(T)
Information table: T

Figure 1: The CURE Framework. The embedding of an object is the concatenation of the embedded vectors of its values.

In addition, there are some embedding methods, e.g., in [Cox and Cox, 2000; Hinton and Roweis, 2002] which optimize the embedding representation on the similarity matrix, but their results heavily rely on the underlying similarity measures. Some other similar embedding methods, e.g. in [Zhang *et al.*, 2015] require class labels to learn distance, and thus they are inapplicable for unsupervised tasks.

Coupling Learning. Coupling learning is a methodology that learns value-to-object coupling relationships to leverage feature and object couplings to empower different models, which has shown valuable and been successfully applied to various problems, e.g., behavior analysis [Cao *et al.*, 2012; Cao, 2014], similarity learning [Wang *et al.*, 2015] and outlier detection [Pang *et al.*, 2016b; 2016a]. This work extends this methodology by capturing hierarchical value to value cluster couplings in categorical data representation.

3 Embedding with Hierarchical Value to Value Cluster Couplings

The proposed CURE framework learns an embedding-based representation for each feature value by modeling hierarchical value to value cluster couplings. As shown in Figure 1, CURE first constructs multiple value-value influence matrices $\{\mathbf{M}_1, \cdots, \mathbf{M}_n\}$ with different value coupling functions $\{\phi_1, \cdots, \phi_n\}$. These value influence matrices reflect the low-level data characteristics. CURE then learns the value clusters with different granularities based on value influence matrices, resulting in a set of value-cluster matrices $\{\mathbf{C}_1, \cdots, \mathbf{C}_n\}$. These value clusters convey rich semantics and have couplings with each other. CURE further learns the couplings between value clusters and generates a $|V| \times r$ value representation matrix $\mathbf{N}$, where r is the dimension in the embedding space. After this, the object embedding matrix is generated by the concatenation of value vectors.

We further instantiate the CURE framework into an embedding method called CDE. In CDE, we construct two value influence matrices to capture the complementary feature value couplings; the complementarity is theoretically proved. CDE learns the value clusters with different granularities by multiple k-means clustering with different parameter values k. By conducting PCA on value clusters, CDE learns the linear correlations between value clusters and obtains the final numerical representation of an original data set.

3.1 Preliminaries

Let a data set $\mathcal{D}$ consist of a number of data objects $\mathcal{O}$ that are described by a set of features $\mathcal{F}$. $\mathcal{D}$ can be organized as an information table $T = <\mathcal{O}, \mathcal{F}, \mathcal{V}>$, where $\mathcal{O} = \{o_1, ..., o_n\}$ is composed of a non-empty finite set of data objects, $\mathcal{F} = \{f_1, ..., f_m\}$ is a finite set of features and $\mathcal{V} = \cup_{j=1}^{m} \mathcal{V}_j$ consists of sets of values from all features, in which $\mathcal{V}_j$ is the set of values of feature f_j. The value sets of each feature are distinct , i.e., $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset, \forall i \neq j$. The whole value set of T is $\mathcal{V} = \{v_1, v_2, ..., v_l\}$, where l is the total number of values.

The value from feature f in object o is denoted by v_o^f and the feature which the value v_i belongs to is denoted by f^i. We assume that the probability $p(v)$ of a value can be represented by its frequency. The joint probability of two values v_i and v_j is $p(v_i, v_j) = \frac{|\{v_o^{f^i}=v_i \cap v_o^{f^j}=v_j, o \in \mathcal{O}\}|}{n}$.

We use normalized mutual information [Estévez *et al.*, 2009] ψ to reflect the relation between two features, which is defined as follows:

$$\psi(f_a, f_b) = \frac{2 \sum\limits_{v_i \in \mathcal{V}_{f_a}} \sum\limits_{v_j \in \mathcal{V}_{f_b}} p(v_i, v_j) log \frac{p(v_i, v_j)}{p(v_i)p(v_j)}}{h(f_a) + h(f_b)}, \quad (1)$$

where $h(f_a) = -\sum_{v_i \in \mathcal{V}_{f_a}} p(v_i) log(p(v_i))$ is the marginal entropy of feature f_a.

3.2 Learning Complementary Value Couplings

We construct two value influence matrices to capture the value couplings from occurrence and co-occurrence perspectives whose complementarity is proved in Section 4.

Definition 1 (Occurrence-based Value Influence Matrix)**.** *The occurrence-based value influence matrix* $\mathbf{M}_o$ *is defined as follows:*

$$\mathbf{M}_o = \begin{bmatrix} \phi_o(v_1, v_1) & \dots & \phi_o(v_1, v_l) \\ \vdots & \ddots & \vdots \\ \phi_o(v_l, v_1) & \dots & \phi_o(v_l, v_l) \end{bmatrix}, \quad (2)$$

where the coupling function $\phi_o(v_i, v_j) = \psi(f^i, f^j) \times \frac{p(v_j)}{p(v_i)}$ *indicates the occurrence influence on value* v_i *from value* v_j*.*

The coupling function ϕ_o captures the difference between the marginal probabilities of values within their own feature. The mutual information which reflects the feature relation is incorporated as weight on value couplings. .

Definition 2 (Co-occurrence-based Value Influence Matrix)**.** *The co-occurrence-based value influence matrix* $\mathbf{M}_c$ *is defined as follows:*

$$\mathbf{M}_c = \begin{bmatrix} \phi_c(v_1, v_1) & \dots & \phi_c(v_1, v_l) \\ \vdots & \ddots & \vdots \\ \phi_c(v_l, v_1) & \dots & \phi_c(v_l, v_l) \end{bmatrix}, \quad (3)$$

where the coupling function $\phi_c(v_i, v_j) = \frac{p(v_i, v_j)}{p(v_i)}$ *indicates the co-occurrence influence on value* v_i *from value* v_j*.*

The coupling function ϕ_c captures the difference between two values by conditional probabilities across different features. Accordingly, $\mathbf{M}_c$ is asymmetric which means the influence on v_i from v_j is different from the influence on v_j from v_i. The ϕ_c value of two values from the same feature always equals 0 since they never co-occur in the same object.

3.3 Clustering Values with Different Granularities

Based on the above matrices, we can learn the value clusters with different granularities which represent different semantics and well reflect the data characteristics. To learn the value clusters with different granularities, here we conduct clustering on the value matrices with different cluster numbers.

We conduct k-means clustering on $\mathbf{M}_o$ with different k, i.e., $\{k_1, k_2, ..., k_o\}$, and on $\mathbf{M}_c$ with $\{k_1, k_2, ..., k_c\}$. The clustering results are represented by a cluster membership indicator matrix, where the entry is one if a value is contained in a value cluster and zero otherwise. So we obtain two indicator matrices. We further concatenate these two indicator matrices and obtain a $l \times (\sum_{i=1}^{o} k_i + \sum_{j=1}^{c} k_j)$ indicator matrix $\mathbf{I}$. The choice of k is discussed in Section 3.5.

k-means clustering is chosen for two major reasons as follows: (1) The value influence matrices are numerical and the Euclidean distance fed in k-means clustering captures the global relation between values. (2) k-means clustering is linear w.r.t. the size of the input matrix, which enables CDE to efficiently learn value clusters with different size.

3.4 Embedding Values with Linear Couplings between Value Clusters

The indicator matrix $\mathbf{I}$ conveys rich couplings between the value clusters obtained using different granularities on two value influence matrices. For simplicity and the consideration of common scenarios, we assume that couplings between value clusters are linear correlations, and apply PCA on the indicator matrix to learn the linear correlations between value clusters to obtain a vector embedding for each value. PCA is chosen because (1) it reduces the data complexity with little loss of information by converting a matrix with linearly correlated variables to a new matrix with linearly uncorrelated components, and (2) it substantially reduces the dimensionality of the value embedding, which enables us to represent an object in a considerably lower-dimensional embedding space.

We first calculate the centralized matrix $\mathbf{X}$ of the indicator matrix $\mathbf{I}$ by subtracting the mean of each column and further derive a covariance matrix $\mathbf{S}$ from $\mathbf{X}$. The value embedding $\mathbf{N}$ is obtained by the following matrix decomposition:

$$\mathbf{N} = \mathbf{X}\mathbf{V}^T, \quad (4)$$

where $\mathbf{V}$ is the principal component matrix derived from singular value decomposition results of $\mathbf{S}$, i.e., $\mathbf{S} = \mathbf{U\Sigma V}$.

After the transformation of PCA, the dimensions of value embedding $\mathbf{N}$ are independent of each other so that the algebraic operations in the Euclidean space can be used on the embedded matrix.

Algorithm 1 *Value Embedding ($\mathcal{D}$, α, β)*

```
Input: $\mathcal{D}$ - data set, $\alpha$ - proportion factor, $\beta$ - dimension reducing factor
Output: $\mathbf{N}$ - the numerical representation of all values
 1: Generate $\mathbf{M}_o$ and $\mathbf{M}_c$
 2: Initialize $\mathbf{I} = \emptyset$
 3: for $\mathbf{M} \in \{\mathbf{M}_o, \mathbf{M}_c\}$ do
 4:     Initialize $k = 2$
 5:     $rm = \emptyset$
 6:     repeat
 7:         $\mathbf{I} = [\mathbf{I}; kmeans(\mathbf{M}, k)]$
 8:         Remove the cluster with only one value and store the remove cluster in $rm$
 9:         $k+ = 1$
10:     until $length(rm) \geq \lceil \frac{k}{\alpha} \rceil$
11: end for
12: $\mathbf{X} = \mathbf{I} - mean(\mathbf{I})$
13: Calculate the covariance matrix $\mathbf{S}$ of $\mathbf{X}$
14: [$\mathbf{U}$, $\mathbf{\Sigma}$, $\mathbf{V}$] = SVD ($\mathbf{S}$)
15: $\mathbf{N} = \mathbf{X}\mathbf{V}^T$
16: Remove the columns whose maximum Euclidean distance of any two elements is less than $\beta$ from $\mathbf{N}$
17: return $\mathbf{N}$
```

3.5 The CDE Method

Algorithm 1 presents the main procedures of CDE. The first step is to generate the value influence matrices $\mathbf{M}_o$ and $\mathbf{M}_c$ according to Definitions (1) and (2).

k is the clustering parameter which decides the granularity of value clusters. Instead of giving a fixed value, we use another proportion factor α to decide the maximum cluster number as shown in Steps (3-11) of Algorithm 1. We remove those tiny clusters with only one value from the indicator matrix. When the number of removed clusters is larger than $\frac{k}{\alpha}$, we stop increasing k, whose initial value is 2. The final indicator matrix is the concatenation of all clustering results with different k from $\mathbf{M}_o$ and $\mathbf{M}_c$.

After conducting PCA on the indicator matrix to learn the correlations between value clusters, we remove those columns whose maximum pairwise Euclidean distance is less than β from $\mathbf{N}$. Finally, we calculate the object embedding by concatenating embedding vectors of its values from $\mathbf{N}$.

We can scan the original data set and generate $\mathbf{M}_o$ and $\mathbf{M}_c$ with the complexity of $O(nm^2)$. Clustering on the value matrix has complexity $O(k_{max}l)$, where k_{max} is the number of times for clustering on one value matrix which is less than value number l. PCA has $O(l^3)$. With the numerical representation of values, generating the embedding matrix of objects has $O(nm)$. Correspondingly, the time complexity of CDE is $O(nm^2 + l^3)$.

4 Theoretical Analysis of CDE

CDE obtains the value clusters by k-means clustering which is based on the Euclidean distance matrices of $\mathbf{M}_o$ and $\mathbf{M}_c$. The distance matrix in k-means clustering decides the quality of value clusters. By proving the complementarity of the two distance matrices, we can observe that the two value couplings are complementary.

The occurrence distance between values v_i and v_j is defined as follows:

$$d_o(v_i, v_j) = \sqrt{\sum_{h=1}^{l} (\phi_o(v_i, v_h) - \phi_o(v_j, v_h))^2}, \qquad (5)$$

where $\phi_o(v_i, v_h)$ is the occurrence coupling function defined in Definition 1 and l is the number of values.

The co-occurrence distance between values v_i and v_j is defined below:

$$d_c(v_i, v_j) = \sqrt{\sum_{h=1}^{l} (\phi_c(v_i, v_h) - \phi_c(v_j, v_h))^2}, \qquad (6)$$

where $\phi_c(v_i, v_h)$ is the co-occurrence coupling function defined in Definition 2. If any two distinct values can be distinguished by d_o or d_c, then d_o and d_c are complementary.

Theorem 1 (Distance Complementarity)**.** *For any two values $v_i \neq v_j$, $d_o(v_i, v_j) \neq 0$ or $d_c(v_i, v_j) \neq 0$.*

Proof. To prove the above theorem, we prove that $v_i \neq v_j$ and $d_o(v_i, v_j) = 0$ satisfy $d_c(v_i, v_j) \neq 0$ for all cases. If $d_c(v_i, v_j) = 0$, then $\forall v_h \in V, \phi_c(v_i, v_h) = \phi_c(v_j, v_h)$. To prove $d_c(v_i, v_j) \neq 0$, we only need to prove $\exists v_h \in V, \phi_c(v_i, v_h) \neq \phi_c(v_j, v_h)$. Then we prove the theorem by considering the following cases.

(1) v_i and v_j belong to the same feature, which means $\psi(f^i, f^h) = \psi(f^j, f^h)$: then $d_o(v_i, v_j) = 0$ if and only if $p(v_i) = p(v_j)$. Let $v_h = v_i$, then $\phi_c(v_i, v_h) = 1$ and $\phi_c(v_j, v_h) = 0$ since v_i, v_j belong to the same feature. Hence, $d_c(v_i, v_j) \neq 0$ when v_i and v_j from the same feature.

(2) v_i and v_j belong to different features: $d_o(v_i, v_j) = 0$ if and only if $\forall v_h \in V, \psi(f^i, f^h)\frac{p(v_h)}{p(v_i)} = \psi(f^j, f^h)\frac{p(v_h)}{p(v_j)}$. When $\psi(f^i, f^h) \neq \psi(f^j, f^h)$ and $p(v_i) \neq p(v_j)$ (suppose $p(v_i) < p(v_j)$), then $p(v_i, v_j) < p(v_j)$. Let $v_h = v_i$, then $\phi_c(v_i, v_h) = 1$ and $\phi_c(v_j, v_h) > 0$. Accordingly, $d_c(v_i, v_j) \neq 0$ when $p(v_i) \neq p(v_j)$. When $\psi(f^i, f^h) = \psi(f^j, f^h)$ and $p(v_i) = p(v_j)$, $\exists v_h$ in feature f^i and $p(v_j, v_h) > 0$, but $p(v_i, v_h) = 0$, then $\phi_c(v_j, v_h) \neq \phi_c(v_i, v_h)$. Therefore, $d_c(v_i, v_j) \neq 0$ when v_i and v_j belong to different features. □

5 Experiments and Evaluation

5.1 Baseline Methods and Parameter Settings

To test the embedding performance, CDE is compared with three popular unsupervised categorical data embedding methods: 0-1 embedding (noted as 0-1), 0-1 embedding with PCA (0-1P), and inverse document frequency embedding (IDF). 0-1 embedding keeps the most complete information in the original data. 0-1 embedding with PCA incorporates feature correlations into the embedding. The IDF embedding differentiates values w.r.t. frequency.

To the best of our knowledge, no existing embedding methods capture the value couplings in categorical data as in CDE.

To test the CDE-based learning performance, we transform CDE to similarity measure and compare it with three typical and well-performed similarity measures which involve feature relation: COS [Wang *et al.*, 2015], DILCA [Ienco *et al.*, 2012] and ALGO [Ahmad and Dey, 2007][1].

In Table 2, $|C|$ is the number of ground-truth classes in data, which is used for the clustering evaluation. We set parameter $\alpha = 10$ in CDE and parameter $\beta = 10^{-10}$ in PCA used by CDE and 0-1P. In COS, DILCA and ALGO, we use the default parameters in their original papers.

5.2 Performance Evaluation Methods

K-means clustering is used to test the performance of CDE against other embedding methods. The embedding methods transform categorical data into numerical data, hence *k*-means clustering can efficiently cluster objects without computing the pairwise object similarity matrix.

To make a fair comparison with similarity-based representation methods, we perform the Gaussian similarity measure on CDE to obtain a object similarity matrix. Spectral clustering is used to evaluate the performance of this object similarity matrix against other object similarity matrices obtained by COS, DILCA and ALGO.

F-score and NMI [Powers, 2011] are two of the most popular clustering evaluation methods. Since we fix the cluster number to the number of classes in each data set, NMI performs similarly as F-score. Here we only report the results of F-score. Higher F-score indicates better clustering accuracy driven by a better embedding method or similarity measure. The p-value results are based on the paired two-tailed t-test using the null hypothesis that the clustering results of CDE and other methods come from distributions with equal means. For each data set, the F-score is the average over 50 validations of clustering with distinct starting points due to the instability of *k*-means clustering.

5.3 Data Sets and Data Factors

We use ten real-world UCI data sets from different domains for the experiments. Various *data factors* are used to measure the underlying characteristics of data sets, which are associated with the learning performance of embedding methods. Two key data factors are defined below, and their results of the data sets are reported in Table 1 and Table 2.

- The *feature correlation index* (FCI) measures the average correlation strength between features:

$$FCI = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i}^{m} SU(f_i, f_j). \qquad (7)$$

  SU measures the correlation between features f_i and f_j by the symmetric uncertainty [Yu and Liu, 2003]. Larger *FCI* indicates stronger correlation between features.

- The *value cluster index* (VCI) is the average of the maximum non-overlapping ratio between value sets contained in different classes for each feature:

$$VCI = \frac{1}{m} \sum_{h=1}^{m} max_{i,j}\{1 - \frac{|\mathcal{V}_{C_i}^h \bigcap \mathcal{V}_{C_j}^h|}{|\mathcal{V}_{C_i}^h \bigcup \mathcal{V}_{C_j}^h|}\}, \qquad (8)$$

  where $\mathcal{V}_{C_i}^h$ is the value set in class C_i for feature f_h and m is the number of features. Larger *VCI* indicates the higher discriminative ability of the value sets.

Table 1: F-score Results of CDE vs. Three Embedding Methods on 10 Data Sets in *k*-means Clustering. Note: The best performance for each data set is boldfaced.

Basic data info. & Data Factor				**F-score**			
Data	$\|O\|$	$\|V\|$	FCI	CDE	0-1	0-1P	IDF
Wisconsin	683	89	0.212	**0.967**	0.946	0.946	0.943
Soybeansmall	47	58	0.180	**0.915**	0.829	0.854	0.763
Mushroom	5644	97	0.148	**0.731**	0.709	0.694	0.506
Mammographic	830	20	0.116	0.809	0.793	**0.815**	0.517
Zoo	101	30	0.110	**0.647**	0.596	0.607	0.537
Dermatology	366	129	0.089	**0.670**	0.598	0.606	0.616
Hepatitis	155	36	0.085	0.680	**0.681**	0.667	0.535
Adult	30162	98	0.060	**0.654**	0.585	0.588	0.479
Lymphography	148	59	0.057	0.418	0.381	0.379	**0.561**
Primarytumor	339	42	0.020	**0.240**	0.230	0.238	0.190
Average				**0.673**	0.635	0.640	0.565
				p-value	0.003	0.003	0.020

5.4 Results and Observations

CDE is first compared with three embedding methods, followed by a comparison with three similarity measures. We then examine the parameter sensitivity of CDE. [2]

Comparison with Embedding Methods

F-score of CDE compared with 0-1, 0-1P and IDF are shown in Table 1. CDE obtains the best F-score on seven data sets; and on average, it demonstrates an approximate 9%, 5% and 19% improvement over 0-1, 0-1P and IDF, respectively. The significance test results show that CDE significantly outperforms other embedding methods at the 95% confidence level.

According to the data factor FCI, the F-score performance of CDE, 0-1 and 0-1P has a downward trend with the decrease of FCI. Since CDE and 0-1P are able to capture the correlation between features according to the data factor FCI, for most data sets with higher FCI, e.g., *Wisconsin, Soybeansmall, Mammographic, Zoo, Dermatology*, CDE outperforms the other embedding methods and 0-1P obtains better performance than 0-1. On the other hand, FCI only reflects the pairwise correlation between features, while CDE captures the couplings beyond such feature correlation. So CDE also performs well on data sets with lower FCI, e.g., *Adult, Primarytumor*. IDF obtains better results on the data sets with weak couplings, especially when the clustering division is consistent with feature-value frequency, e.g., *Lymphography*.

Comparison with Similarity Measures

The CDE-based Gaussian similarity (denoted by CDE-G) is compared with three well-performing similarity measures:

[1] Our experiments show DM [Jia *et al.*, 2016] underperforms DILCA and ALGO, so its results are thus omitted due to space limit.

[2] CDE runs one order of magnitude slower than other embedding methods and much faster than other similarity measures. Due to space limitations, we do not show the detailed efficiency results here.

Table 2: F-score Results of CDE-G vs. Three Coupled Similarity Measures on 10 Data Sets in Spectral Clustering. Note: COS, DILCA and ALGO run out of memory on *Adult*. The average values are computed according to first nine data sets.

Clustering Info & Data Factor			**F-score**			
Data	$\|C\|$	VCI	CDE-G	COS	DILCA	ALGO
Primarytumor	21	0.873	**0.242**	0.196	0.224	0.209
Zoo	7	0.733	**0.644**	0.538	0.583	0.547
Soybeansmall	4	0.712	**1.000**	0.893	0.910	0.911
Lymphography	4	0.699	**0.397**	0.395	0.353	0.366
Dermatology	6	0.664	0.784	0.730	**0.808**	0.710
Mushroom	2	0.310	**0.828**	0.825	0.826	0.826
Wisconsin	2	0.237	0.962	**0.973**	0.921	0.971
Hepatitis	2	0.141	0.667	0.463	**0.679**	0.662
Mammographic	2	0.071	0.817	**0.828**	0.826	0.818
Adult	2	0.032	0.676	NA	NA	NA
Average			**0.762**	0.706	0.738	0.726
			p-value	0.050	0.100	0.032

COS, DILCA and ALGO. As shown in Table 2, CDE-G remains the best performer on half of the data sets. CDE-G obtains about 8%, 3% and 5% improvement over COS, DILCA and ALGO respectively in terms of F-score. The significance test results show that CDE-G significantly outperforms the other similarity measures at 90% confidence level. Note that COS, DILCA and ALGO on *Adult* run out of memory since calculation of object similarity needs a large amount of memory. This shows that it is more efficient to represent categorical data with an embedding matrix than a similarity matrix.

In Table 2, the data sets are sorted in the descending order of VCI which reflects the discriminative ability of the value clusters in object classes. The class number $|C|$ is also a factor to describe the complexity of data clustering, which is consistent with VCI according to Table 2. Since CDE-G makes use of the value clusters with different granularities, on most data sets with larger VCI and larger $|C|$, CDE-G achieves better performance than the other similarity measures. Since CDE-G, COS, DILCA and ALGO are able to capture the pairwise correlation between features, all methods achieve good performance on data sets with higher FCI.

Sensitivity Test w.r.t. Parameters α and β

There are two parameters in CDE: α controls the dimension of value embedding before PCA and β controls the dimension of value embedding after PCA. Since all results have a similar trend, we demonstrate the results of four data sets: *Adult, Dermatology, Wisconsin, Primarytumor*, which have the largest $|O|$, $|V|$, FCI and VCI respectively.

Figure 2 shows the dimension of value embedding before PCA and the clustering performance with different α. α directly influences the value of k in Algorithm 1. k determines the granularity of value clusters which consist of the original value embedding. Since we only drop the clusters with only one value, the clustering performance is stable with parameter α, and we can choose the parameter value which is associated with the low dimension of embedding. According to Figure 2, the dimension is stable when $\alpha \geq 10$.

Figure 3 shows the dimension of the final value embedding and the clustering performance w.r.t. β.

It shows that the performance of the clustering is stable with β. When $\beta \geq 10^{-15}$, the dimension of value embedding vectors decreases with the increase of β on all data sets.

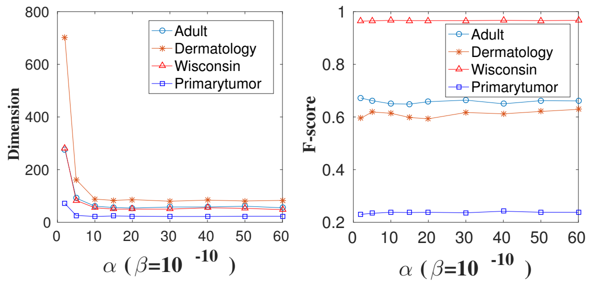

Figure 2: Sensitivity Test of Parameter α on Four Data Sets.

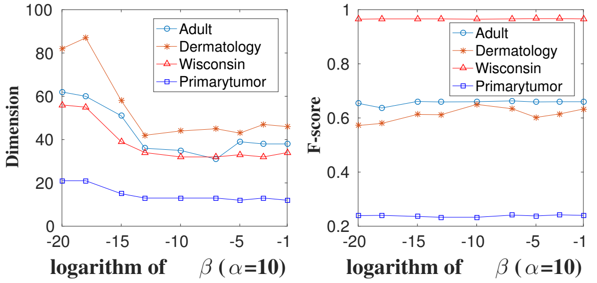

Figure 3: Sensitivity Test of Parameter β on Four Data Sets.

According to Figures 2 and 3, the clustering performance is not sensitive to parameters α and β. These two parameters can influence the dimension of value embedding. The dimension is stable when $\alpha \geq 10$ and $\beta \geq 10^{-15}$.

6 Conclusions

Different from existing encoding-based embedding and feature correlation-based similarity measures, a novel unsupervised representation framework (CURE) and its instantiation (CDE) are introduced in this paper, which model hierarchical value couplings in terms of feature interactions and value clustering. Extensive experiments show that CDE significantly outperforms typical embedding methods and similarity measures in capturing feature value interactions. In addition, two proposed data factors further indicate the feature value couplings and value clusters in data sets. Our future work is to model selective value couplings and instantiate the framework into other instances to suit different applications.

Acknowledgments

This work is partially supported by The National Key Research and Development Program of China (No.2016YFB0200401), by program for New Century Excellent Talents in University, by NSF China 61402492, 61402486, 61379146, by the laboratory pre-reseach fund (9140C810106150C81001) and by the Australian Research Council Discovery Grant (DP130102691).

References

[Ahmad and Dey, 2007] Amir Ahmad and Lipika Dey. A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63(2):503–527, 2007.

[Aizawa, 2003] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[Bengio *et al.*, 2013] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

[Cao *et al.*, 2012] Longbing Cao, Yuming Ou, and S Yu Philip. Coupled behavior analysis with applications. *IEEE Transactions on Knowledge and Data Engineering*, 24(8):1378–1392, 2012.

[Cao, 2014] Longbing Cao. Non-iidness learning in behavioral and social data. *The Computer Journal*, 57(9):1358–1370, 2014.

[Cao, 2015] Longbing Cao. Coupling learning of complex interactions. *Information Processing & Management*, 51(2):167–186, 2015.

[Cohen *et al.*, 2013] Jacob Cohen, Patricia Cohen, Stephen G West, and Leona S Aiken. *Applied multiple regression/correlation analysis for the behavioral sciences*. Routledge, 2013.

[Cox and Cox, 2000] Trevor F Cox and Michael AA Cox. *Multidimensional Scaling*. CRC press, 2000.

[Deerwester *et al.*, 1990] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391, 1990.

[Estévez *et al.*, 2009] Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.

[Foss and Zaïane, 2002] Andrew Foss and Osmar R Zaïane. A parameterless method for efficiently discovering clusters of arbitrary shape in large datasets. In *Proceedings of ICDM*, pages 179–186. IEEE, 2002.

[Hinton and Roweis, 2002] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Proceedings of NIPS*, pages 833–840, 2002.

[Hofmann, 1999] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of SIGIR*, pages 50–57. ACM, 1999.

[Ienco *et al.*, 2012] Dino Ienco, Ruggero G Pensa, and Rosa Meo. From context to distance: Learning dissimilarity for categorical data clustering. *ACM Transactions on Knowledge Discovery from Data*, 6(1):1, 2012.

[Jia *et al.*, 2016] Hong Jia, Yiu-ming Cheung, and Jiming Liu. A new distance metric for unsupervised learning of categorical data. *IEEE Transactions on Neural Networks and Learning Systems*, 27(5):1065–1079, 2016.

[Jolliffe, 2002] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119, 2013.

[Pang *et al.*, 2016a] Guansong Pang, Longbing Cao, and Ling Chen. Outlier detection in complex categorical data by modelling the feature value couplings. In *Proceedings of IJCAI*, pages 1902–1908. AAAI, 2016.

[Pang *et al.*, 2016b] Guansong Pang, Longbing Cao, Ling Chen, and Huan Liu. Unsupervised feature selection for outlier detection by modelling hierarchical value-feature couplings. In *Proceedings of ICDM*, pages 410–419. IEEE, 2016.

[Pang *et al.*, 2016c] Guansong Pang, Kai Ming Ting, David Albrecht, and Huidong Jin. Zero++: Harnessing the power of zero appearances to detect anomalies in large-scale data sets. *Journal of Artificial Intelligence Research*, 57:593–620, 2016.

[Powers, 2011] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[Wang *et al.*, 2015] Can Wang, Xiangjun Dong, Fei Zhou, Longbing Cao, and Chi-Hung Chi. Coupled attribute similarity learning on categorical data. *IEEE Transactions on Neural Networks and Learning Systems*, 26(4):781–797, 2015.

[Wilson and Chew, 2010] Andrew T Wilson and Peter A Chew. Term weighting schemes for latent dirichlet allocation. In *The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 465–473. Association for Computational Linguistics, 2010.

[Yu and Liu, 2003] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of ICML*, volume 3, pages 856–863, 2003.

[Zhang *et al.*, 2015] Kai Zhang, Qiaojun Wang, Zhengzhang Chen, Ivan Marsic, Vipin Kumar, Guofei Jiang, and Jie Zhang. From categorical to numerical: Multiple transitive distance learning and embedding. In *Proceedings of SDM*. SIAM, 2015.