The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

An Euclidean Distance Based on Tensor Product Graph Diffusion Related Attribute Value Embedding for Nominal Data Clustering

Lei Gu,[1] **Ningning Zhou,**[1] **Yang Zhao**[2]
[1] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China
[2] College of Humanities and Social Sciences, Nanjing Forestry University, Nanjing, China
leon_gu@yeah.net, zhounn@njupt.edu.cn, zwx_zy@njfu.edu.cn

Abstract

Not like numerical data clustering, nominal data clustering is a very difficult problem because there exists no natural relative ordering between nominal attribute values. This paper mainly aims to make the Euclidean distance measure appropriate to nominal data clustering, and the core idea is the attribute value embedding, namely, transforming each nominal attribute value into a numerical vector. This embedding method consists of four steps. In the first step, the weights, which can quantify the amount of information in attribute values, is calculated for each value in each nominal attribute based on each object and its k nearest neighbors. In the second step, an intra-attribute value similarity matrix is created for each nominal attribute by using the attribute value's weights. In the third step, for each nominal attribute, we find another attribute with the maximal dependence on it, and build an inter-attribute value similarity matrix on the basis of the attribute value's weights related to these two attributes. In the last step, a diffusion matrix of each nominal attribute is constructed by the tensor product graph diffusion process, and this step can cause the acquired value embedding to contain simultaneously the intra- and inter-attribute value similarities information. To evaluate the effectiveness of our proposed method, experiments are done on 10 data sets. Experimental results demonstrate that our method not only enables the Euclidean distance to be used for nominal data clustering, but also can acquire the better clustering performance than several existing state-of-the-art approaches.

Introduction

Most of data mining techniques are only applicable to numerical data and can not operate well for nominal data consisting of several nominal attributes which have no numerical values, such as degree or profession, since nominal attributes have unordered scales and mathematical calculations, like addition or subtraction, are unable to be carried out on them(Agresti 2007). To make nominal data clustering practicable, nowadays, there exist two primary kinds of methods: designing the specific distance metrics to quantify the dissimilarities between nominal attribute values and transforming nominal values into embedding vectors .

The most straightforward and generally used specific distance metric for nominal values is the Hamming distance (HAM for short)(Bock 2000). Additionally, with increasing efforts of researchers, more latest special distance metrics are proposed for nominal values in recent years, such as the coupled nominal distance (CNS)(Wang et al. 2015; 2011), the Hong's distance metric (HDM)(Jia, m. Cheung, and Liu 2016) and the Ahmad's distance metric (ADM)(Ahmad and Dey 2007). Nevertheless, these distance metrics have some disadvantages. HAM is very simple, but it does not notice the dependence between attributes; HDM focus on the combination of two attributes, but it ignores the differences between multiple attributes; CNS and ADM all take into consideration that there may be certain relations between two values from one attribute with respect to other attributes, but they fail to give proper attention to the correlations between values from distinct attributes and are also ineffective for nominal data with totally independent attributes. In addition, when implementing the clustering task, we should incorporate the aforementioned distance metrics into the models of some nominal clustering algorithms (e.g., the most popular K-modes(Huang 1998)), but the number of nominal clustering algorithms is far less than that of numerical clustering algorithms.

The embedding representation methods can convert nominal values into numerical vectors. Accordingly, after this transformation, the numerical distance metrics (e.g., the simple Euclidean distance) can be inserted into a lot of effective numerical clustering algorithms (e.g., the most popular K-means) and be used to execute clustering on the changed nominal data. The available and classical value embedding methods for nominal data clustering are few now. Nevertheless, we can still roughly summarize three existing value embedding methods: the dummy variables related value embedding (DVE)(Suits 1957; Zdravevski et al. 2015), the term frequency–inverse document frequency related value embedding (TVE) and the coupled data embedding (CDE)(Songlei Jian 2017). Nonetheless, there are also obvious drawbacks in these three approaches. First, DVE only can transform each nominal attribute value into a one dimensional vector, such as a number 0 or 1, and it overlooks absolutely any relevance contained within nominal data. Second, the term frequency–inverse document frequency(TF-IDF)(Berry 2003) is often applied to document analysis, but whether it is appropriate to general nominal data still requires the support of the theory foundation.

Therefore, TVE as a general value embedding method has not yet widely used until now. Third, although, in CDE, each nominal value can be changed into a vector made up of multiple numbers and the dependency degree between values is also measured, it is very easy for CDE to generate the high dimensional data.

To spare the trouble of selecting only from a small amount of nominal data clustering algorithms utilizing the specific distance metrics, in this paper, we endeavor to validate the Euclidean distance embedded into the frameworks of numerical clustering approaches for nominal data and propose a tensor product graph diffusion related attribute value embedding method (TAVE), which is made up of four steps: calculating attribute value's weights, creating intra-attribute value similarity matrices, building inter-attribute value similarity matrices and constructing diffusion matrices. The presented TAVE has the five following characteristics: (1) In comparison to all existing methods mentioned above, TAVE not only can capture the intrinsic information and relations from each nominal object and its k nearest neighbors, but also can utilize the tensor product graph diffusion process to propagate the intra- and inter-attribute value similarities and make the obtained numerical vectors hold concurrently these two sorts of similarities information. (2) Contrasted with HAM, CNS, HDM and ADM, the Euclidean distance based on TAVE, which cooperates with the most popular K-means belonging to one type of numerical clustering algorithms, can be utilized to perform clustering on the transformed nominal data. (3) Compared severally with DVE and TVE, TAVE is grounded on information theories and graph diffusion and it is more suitable for general nominal data rather than special data (e.g., documents). (4) TAVE can bring about low dimensional numerical data as against CDE. (5) substantial experiments on benchmark data sets show the higher effectiveness of TAVE in comparison with these existing approaches.

Related Work

In this section, we present an overview of the existing related work from two aspects.

The first aspect is in concern of the dissimilarity metrics measuring the differences between nominal attribute values or objects. This aspect can be divided into three parts again. First, the dissimilarity degree is only related to a single attribute. The most heavily used HAM is one of typical representatives, and HAM between two nominal objects is equal to the number of their mismatched attribute values. Furthermore, some similarity metrics(dos Santos and Zrate 2015; Boriah, Chandola, and Kumar 2008), which are frequently used in nominal data, also belong to this category, such as Gower similarity, Eskin similarity, Lin similarity and Smirnov similarity. Second, the dissimilarity measure involve two attributes, but they only can be employed to express the relationships between two values from one same attribute. One proposed association-based distance(Le and Ho 2005) and ADM are all attributed to this kind of dissimilarity measures; they can measure the dissimilarity between two values from one same attribute with respect to all other different attributes. However, they neglect the intra-attribute dissimilarity between two values from one same attribute. The recently proposed CNS solves this problem. It defines the intra-attribute similarity between two values from one attribute as intra-coupled attribute value similarity, and also defines the similarity between two values from one attribute with respect to all other attributes as inter-coupled attribute value similarity, and then regards the product of two decreasing functions, whose variables are separately the intra-coupled and inter-coupled attribute value similarity, as the final dissimilarity also called coupled attribute value dissimilarity. The key idea of CNS has been successfully applied to dealing with complex tasks, such as matrix factorization(Li, Xu, and Cao 2015), collaborative filtering(Jiang et al. 2015), multi-label classification(Liu and Cao 2015) and outlier detection(Pang, Cao, and Chen 2016). Third, the dissimilarity measure is based on the combinations of only two nominal attributes. The latest work HDM pertains to this kind, and it uses the mutual information to decide which two attributes should be combined.

The second aspect is in regards to the nominal attribute value embedding. This aspect also can be separated into two components again. Firstly, each nominal attribute value is changed into a numerical vector with one dimension. DVE and TVE are all be subordinate to this class. In DVE, a nominal attribute value can be thought of as a dummy variable represented as a numerical value 0 or 1. Dummy variables are also known as indicator variables and are involved frequently in studies of economic forecasting, credit scoring, etc. In TVE, a nominal attribute value is substituted by TF-IDF which is the product of two statistics, term frequency and inverse document frequency. TF-IDF has already been used successfully for document summarization and text classification. Secondly, each nominal attribute value is converted into a numerical vector with multiple dimensions. While this kind of methods is few, the latest CDE is one of them. CDE builds two value matrices to capture the attribute value couplings from occurrence and co-occurrence perspective, learns the value clusters with different granularities based on two value matrices, concatenate the indicator matrices related to the value clusters, and then apply principal component analysis on the final matrix to obtain a vector embedding for each value. Moreover, our proposed TAVE also belongs to this category.

Table 1: An example of nominal data set

X	a_1	a_2	a_3	a_4
x_1	l_1	c_2	g_1	b_1
x_2	l_1	c_3	g_2	b_2
x_3	l_1	c_1	g_1	b_2
x_4	l_1	c_1	g_1	b_2
x_5	l_2	c_1	g_2	b_2
x_6	l_1	c_1	g_2	b_1

Methodology

Assume a nominal data set can be formally described as follows: $X=\{x_1, x_2, \ldots, x_n\}$ is a set of n objects, represented by a set of nominal attributes $A=\{a_1, a_2, \ldots, a_t\}$, where t is the number of attributes; $f(a_m)$ and $\widetilde{a}_m^{\gamma}$ are the number of values and the γth value for attribute a_m respectively (Note that here the attribute value's order have no concern with any kind of attribute value's magnitudes and they only have an effect on differentiating between different values in attribute a_m; $\gamma \in \{1, 2, \ldots, f(a_m)\}$.). Again, let $h(x_i, a_m)$ be the nominal value of attribute a_m for object x_i and $\vec{v}(x_i, a_m)$ be the embedding-based representation of $h(x_i, a_m)$. Each $h(x_i, a_m)$ has no any natural order, but each $\vec{v}(x_i, a_m)$ is a numerical vector. Figure 1 shows the simple flowchart of our TAVE. The proposed TAVE, which is also summarized by Algorithm 1, mainly aims to transform each $h(x_i, a_m)$ in a nominal data set X into a corresponding $\vec{v}(x_i, a_m)$ and make the Euclidean distance between nominal objects work well for nominal data clustering. Table 1 exhibits a nominal data set consisting of six objects and four attributes. Take $h(x_1, a_3)$ as an example, $h(x_1, a_3) = \widetilde{a}_3^1 = g_1$, and when the parameter k=2 and q=10 in Algorithm 1, TAVE can convert this nominal value into $\vec{v}(x_1, a_3)$, that is, a numerical vector $(4.1135, 2.9491, 2.8852, 1.3772)^T$. There are four important steps in Algorithm 1 and we will describe them in the following subsections.

Furthermore, here a new presented Euclidean distance $d(x_i, x_j)$ between two nominal objects x_i and x_j can be defined as:

$$d(x_i, x_j) = \|\vec{e}(x_i) - \vec{e}(x_j)\|_2 \qquad (1)$$

where two vectors $\vec{e}(x_i)$ and $\vec{e}(x_j)$ are separately formed by concatenating all $\vec{v}(x_i, a_m)$ and $\vec{v}(x_j, a_m)$ (m=1, 2,. . . ,t), and $\|\cdot\|_2$ is the L^2 norm.

Algorithm 1 The Proposed TAVE method

```
Input: two parameters $k$ and $q$ ($k, q > 0$), a nominal data set
       $X$, and an attribute value $h(x_i, a_m)$ (here suppose
       $h(x_i, a_m) = \widetilde{a}_m^{\gamma}$; $i \in \{1, 2, \ldots, n\}$, $m \in \{1, 2, \ldots, t\}$,
       and $\gamma \in \{1, 2, \ldots, f(a_m)\}$.)
Output: an embedding vector $\vec{v}(x_i, a_m)$ of $h(x_i, a_m)$

For each nominal attribute value $\widetilde{a}_m^{\beta}$ in attribute $a_m$ ($\beta = 1, 2, \ldots, f(a_m)$), calculate its weight $w(\widetilde{a}_m^{\beta})$ in accordance with its occurrence times within attribute $a_m$ of all objects and their $k$ nearest neighbors by Eq.(2).
For nominal attribute $a_m$, create an intra-attribute value similarity matrix $M(a_m)$ based on the weights $w(\widetilde{a}_m^{\beta})$ of all attribute values in $a_m$ by Eq.(3).
For nominal attribute $a_m$, find a nominal attribute $a_u$ in $A$ ($u \in \{1, 2, \ldots, t\}$, but $u \neq m$) with the maximal dependence on $a_m$, compute the weights $w(\widetilde{a}_u^{\delta})$ ($\delta = 1, 2, \ldots, f(a_u)$) of all nominal attribute values in $a_u$ by Eq.(2), and then build an inter-attribute value similarity matrix $Q(a_m)$ according to the weights $w(\widetilde{a}_m^{\beta})$ and $w(\widetilde{a}_u^{\delta})$ of all attribute values in $a_m$ and $a_u$ by Eq.(4).
For nominal attribute $a_u$, create an intra-attribute value similarity matrix $M(a_u)$ by Eq.(3), and for nominal attribute $a_m$, construct a diffusion matrix $H(a_m)$ on the basis of $M(a_m)$, $M(a_u)$ and $Q(a_m)$ by iterating Eq.(5) until the maximum iteration number $q$ is reached.
Let the $\gamma$th row of $H(a_m)$ be $\vec{v}(x_i, a_m)$.
```

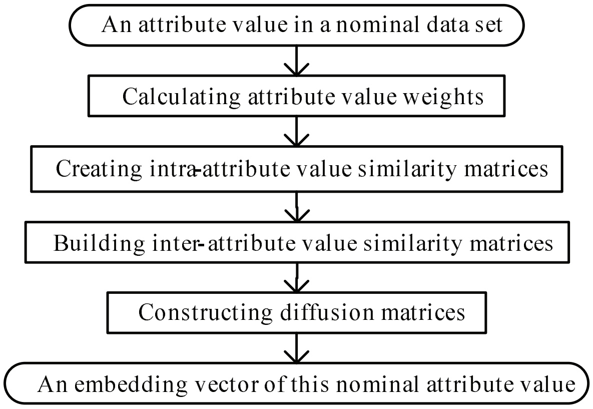
An attribute value in a nominal data set
Calculating attribute value weights
Creating intra-attribute value similarity matrices
Building inter-attribute value similarity matrices
Constructing diffusion matrices
An embedding vector of this nominal attribute value

Figure 1: The simple flowchart of TAVE

Calculating attribute value's weights

Attribute value's weights can be applied to revealing the information content of attribute values, and thus they are considered as the root of TAVE. The weight of each nominal attribute value is calculated by using this value occurrence times in objects. The definition of one attribute value's weight is given as follows:

Definition 1. [Attribute value's weight (w)]
Given a nominal data set X and a value $\widetilde{a}_m^{\beta}$ in attribute a_m. The weight $w(\widetilde{a}_m^{\beta})$ of attribute value $\widetilde{a}_m^{\beta}$ is defined as:

$$w(\widetilde{a}_m^{\beta}) = \frac{\sum_{i=1}^{n} \phi(x_i, \widetilde{a}_m^{\beta}) + \sum_{i=1}^{n} \sum_{\sigma=1}^{k} \phi(\hat{x}_i^{\sigma}, \widetilde{a}_m^{\beta})}{n + n \cdot k} \qquad (2)$$

where k is a parameter, ϕ is a function, and $\hat{x}_i^{\sigma}$ is a object ($\hat{x}_i^{\sigma} \in X$) and the σth nearest neighbor of object x_i.

In Eq.(2), $\phi(\cdot)$ is an indicator function, $\phi(x_i, \widetilde{a}_m^{\beta})$=1 if $\widetilde{a}_m^{\beta}$ is the value of attribute a_m for object x_i and 0 otherwise, and while $\sum_{i=1}^{n} \phi(x_i, \widetilde{a}_m^{\beta})$ and $\sum_{i=1}^{n} \sum_{\sigma=1}^{k} \phi(\hat{x}_i^{\sigma}, \widetilde{a}_m^{\beta})$ are all related to attribute value $\widetilde{a}_m^{\beta}$, they respectively signify the value occurrence times within attribute a_m of n objects x_i and their corresponding k nearest neighbors $\hat{x}_i^{\sigma}$. Moreover, for any attribute value $\widetilde{a}_m^{\beta}$, $w(\widetilde{a}_m^{\beta}) \in [0, 1]$, and for each object, we employ Hamming distance to seek its k nearest neighbors before Eq.(2) is available in TAVE.

For a nominal data set, the weight of each nominal attribute value can be reckoned by Eq.(2). For instance, in Table 1, $h(x_1, a_3) = \widetilde{a}_3^1 = g_1$, and when k=2, $w(\widetilde{a}_3^1)$=0.7222. Two nearest neighbors of x_1 are x_3 and x_4, that is, $\hat{x}_1^1 = x_3$ and $\hat{x}_1^2 = x_4$.

Creating intra-attribute value similarity matrices

Each intra-attribute value similarity matrix can clearly express the relationship between different nominal values in each attribute. Therefore, these similarity matrices are an important part of TAVE. The intra-attribute value similarity matrix of each attribute is created by utilizing all value's weights in this attribute. The related definition is provided as follows:

Definition 2. [Intra-attribute value similarity matrix(M)] *Given a nominal data set X and an attribute a_m in A. The intra-attribute value similarity matrix $M(a_m)$ of attribute a_m is defined as:*

$$\ddot{M}(\rho, \eta \,|\, a_m) = \frac{\min\langle w(\widetilde{a}_m^{\rho}), w(\widetilde{a}_m^{\eta})\rangle}{\max\langle w(\widetilde{a}_m^{\rho}), w(\widetilde{a}_m^{\eta})\rangle} \tag{3}$$

where $\ddot{M}(\rho, \eta \,|\, a_m)$ is the element of matrix $M(a_m)$, ρ and η=$1, 2, \ldots, f(a_m)$ here, min$\langle\cdot\rangle$ and max$\langle\cdot\rangle$ severally return the largest and smallest value from the numbers enclosed in angle brackets, and $w(\cdot)$ is the value's weight.

Each element $\ddot{M}(\rho, \eta \,|\, a_m)$ of matrix $M(a_m)$ computed by using a couple of the attribute value's weights and Eq.(3) can reflect the similarity degree between a pair of attribute values. The larger $\ddot{M}(\rho, \eta \,|\, a_m)$ indicates the higher similarity degree. If the weights of two values (e.g., $\widetilde{a}_m^{\rho}$ and $\widetilde{a}_m^{\eta}$) are same, then we can acquire $\ddot{M}(\rho, \eta \,|\, a_m)$=1. In addition, here one supplementary explanation about Eq.(3) is that $\ddot{M}(\rho, \eta \,|\, a_m)$=1 if $w(\widetilde{a}_m^{\rho})$=$w(\widetilde{a}_m^{\eta})$= 0, and similarly, $\ddot{M}(\rho, \eta \,|\, a_m)$=1 if ρ=η.

At last, for example, for attribute a_3 in Table 1, we can further get $w(\widetilde{a}_3^2)$=0.2778 by Eq.(2) when k=2, and consequently the similarity matrix $M(a_3)$ can be formed based on $w(\widetilde{a}_3^1)$ and $w(\widetilde{a}_3^2)$, that is, $M(a_3)$=$\begin{bmatrix}1.0000 & 0.3846\\ 0.3846 & 1.0000\end{bmatrix}$.

Building inter-attribute value similarity matrices

The intra-attribute value similarity matrices are not fully competent for depicting the characteristics of nominal attribute values. Accordingly, the inter-attribute value similarity matrices are the powerful supplement for capturing more intrinsic relations between nominal attribute values, and they are also regarded as an indispensable component of TAVE. The corresponding definition is shown as follows:

Definition 3. [Inter-attribute value similarity matrix(Q)] *Given a nominal data set X and an attribute a_m in A. The inter-attribute value similarity matrix $Q(a_m)$ of attribute a_m is defined as:*

$$\ddot{Q}(\theta, \lambda \,|\, a_m) = \frac{s(\theta, \lambda)}{n} \cdot \frac{\min\langle w(\widetilde{a}_m^{\theta}), w(\widetilde{a}_u^{\lambda})\rangle}{\max\langle w(\widetilde{a}_m^{\theta}), w(\widetilde{a}_u^{\lambda})\rangle} \tag{4}$$

where $w(\cdot)$, min$\langle\cdot\rangle$ and max$\langle\cdot\rangle$ are identical to the forementioned Eq.(3), $\ddot{Q}(\theta, \lambda \,|\, a_m)$ is the element of matrix $Q(a_m)$, θ=$1, 2, \ldots, f(a_m)$, λ=$1, 2, \ldots, f(a_u)$, a_u (u≠m) is a specially appointed nominal attribute in A, and $s(\theta, \lambda)$ is the co-occurrence times of the attribute values $\widetilde{a}_m^{\theta}$ and $\widetilde{a}_u^{\lambda}$ within a_m and a_u of n objects.

Like $\ddot{M}(\rho, \eta \,|\, a_m)$, each element $\ddot{Q}(\theta, \lambda \,|\, a_m)$ of matrix $Q(a_m)$ is also reckoned by a pair of the nominal attribute value's weights in Eq.(4). However, $\ddot{Q}(\theta, \lambda \,|\, a_m)$ need have the ability to denote the dependence degree between two values $\widetilde{a}_m^{\theta}$ and $\widetilde{a}_u^{\lambda}$ from two different attributes a_m and a_u, and the bigger $\ddot{Q}(\theta, \lambda \,|\, a_m)$ is, the greater the dependence degree is. Hence, an attribute a_u in A should be obtained and its value's weight can be calculated by Eq.(2) before we make use of Eq.(4). Here we select an attribute, which has the highest value of normalized mutual information(Cai, He, and Han 2005) with attribute a_m, from all attributes in A except a_m as the desired attribute a_u. Furthermore, in Eq.(4), $\ddot{Q}(\theta, \lambda \,|\, a_m)$=$\frac{s(\theta,\lambda)}{n}$ if $w(\widetilde{a}_m^{\theta})$=$w(\widetilde{a}_u^{\lambda})$=0.

Finally, take also attribute a_3 in Table 1 as an instance, nominal attribute a_u=a_1 because a_3 has the largest value of normalized mutual information with a_1 in comparison with other attributes, and we can ulteriorly gain $s(1, 1)$=3, $s(1, 2)$=0, $s(2, 1)$=2, $s(2, 2)$=1, and the value's weights $w(\widetilde{a}_1^1)$=0.9444 and $w(\widetilde{a}_1^2)$=0.0556. Therefore, the similarity matrix $Q(a_3)$=$\begin{bmatrix}0.3824 & 0.0000\\ 0.0980 & 0.0333\end{bmatrix}$.

Constructing diffusion matrices

The purpose of constructing diffusion matrices is to fuse the intra- and inter-attribute value similarity matrices, which can represent the information content contained the attribute values from different views. Consequently, this procedure plays a vital role in TAVE. The correlative definition is furnished as follows:

Definition 4. [Diffusion matrix(H)] *Given a nominal data set X and an attribute a_m in A. The diffusion matrix $H(a_m)$ of attribute a_m is constructed by iterating Eq.(5) until the maximum iteration number q is met, that is, ξ=q, and Eq.(5) is defined as(Yang, Prasad, and Latecki 2013; Shu and Latecki 2015):*

$$F^{(\xi+1)} = S\ F^{(\xi)}\ S^T + I \tag{5}$$

where S=$\begin{bmatrix}M(a_m) & Q(a_m)\\ Q(a_m)^T & M(a_u)\end{bmatrix}$ and I is the identity matrix. Here let $H(a_m)$=$F^{(q+1)}$ and $F^{(1)}$=S.

To acquire the diffusion matrix of attribute a_m, we should produce the matrices $M(a_m)$, $M(a_u)$ and $Q(a_m)$ in advance, and then implement the iteration of Eq.(5). The theoretical analysis of Eq.(5) is detailed in the next section.

Lastly, take still nominal attribute a_3 in Table 1 for example, according to the matrices $M(a_3)$, $Q(a_3)$ and $M(a_1)$ ($M(a_1)$=$\begin{bmatrix}1.0000 & 0.0588\\ 0.0588 & 1.0000\end{bmatrix}$), when the iteration number q=10, the diffusion matrix $H(a_3)$ can be given as follows:

$$H(a_3) = \begin{bmatrix}4.1135 & 2.9491 & 2.8852 & 1.3772\\ 2.9492 & 4.2132 & 2.6159 & 1.4686\\ 2.8855 & 2.6161 & 4.0981 & 1.6403\\ 1.3804 & 1.4718 & 1.6433 & 6.0010\end{bmatrix}.$$

It note that the first row vector of $H(a_3)$ is the embedding vector $\vec{v}(x_1, a_3)$ of nominal attribute value $\widetilde{a}_3^1$(namely, g_1 or $h(x_1, a_3)$).

Theoretical Analysis

Firstly, in most of nominal data clustering methods, the occurrence frequency of each attribute value is often used as the similarity or dissimilarity measure. Nevertheless, according to probability theory(Soong 2004), this kind of the occurrence frequencies only can be called relative likelihood because it indeed become the real frequency also named relative frequency only when the number of objects should be enough large. Therefore, we view each object as a prototype and treat its k nearest neighbors as the augmented objects to apply them for more clearly depicting the frequency feature of each attribute value, and these nearest neighbors are derived from the original data set and are most similar to their prototypes. The Eq.(2) is devised in terms of the original and augmented objects.

Theorem 1 *For all nominal values* $\widetilde{a}_m^{\varepsilon}$ $(\varepsilon{=}1,2,\ldots,f(a_m)$ *in attribute* a_m*,* $\sum\limits_{\varepsilon=1}^{f(a_m)} w(\widetilde{a}_m^{\varepsilon}){=}1$.

Proof. Obviously, $\frac{1}{n}\sum\limits_{\varepsilon=1}^{f(a_m)}\sum\limits_{i=1}^{n}\phi(x_i,\widetilde{a}_m^{\varepsilon}){=}1$ according to the indicator function $\phi(\cdot)$ of Eq.(2) because $\sum\limits_{i=1}^{n}\phi(x_i,\widetilde{a}_m^{\varepsilon})$ and $\frac{1}{n}\sum\limits_{i=1}^{n}\phi(x_i,\widetilde{a}_m^{\varepsilon})$ are separately the occurrence times and frequency of attribute value $\widetilde{a}_m^{\varepsilon}$ within attribute a_m of n original objects. Since each original object has k nearest neighbors regarded as the augmented objects, the size of the original and augmented objects add up to $n{+}n{\cdot}k$. Consequently, we can obtain $\frac{1}{n+n\cdot k}\sum\limits_{\varepsilon=1}^{f(a_m)}\sum\limits_{i=1}^{n+n\cdot k}\phi(\bar{x}_i,\widetilde{a}_m^{\varepsilon}){=}1$ on $n{+}n{\cdot}k$ objects. Here, object $\bar{x}_i$ comes from n original and $n{\cdot}k$ augmented objects. Since we can get $w(\widetilde{a}_m^{\varepsilon}){=}\frac{1}{n+n\cdot k}\sum\limits_{i=1}^{n+n\cdot k}\phi(\bar{x}_i,\widetilde{a}_m^{\varepsilon})$ based on Eq.(2), $\sum\limits_{\varepsilon=1}^{f(a_m)} w(\widetilde{a}_m^{\varepsilon}){=}1$.

Secondly, in Eq.(5), if all values in attribute a_m and a_u are considered as all vertices of a graph N, S is the adjacency matrix of graph N and also called the affinity matrix. For Eq.(5), assume that we symbolize the limit matrix by $F^{*}{=}\lim_{\xi\to\infty}F^{(\xi)}$; furthermore, again suppose that a tensor product graph $\mathbb{N}{=}N{\otimes}N$ where $\otimes$ is the Kronecker product, and $\mathbb{S}(\mathbb{S}{=}S{\otimes}S)$ is the adjacency matrix of a tensor product graph $\mathbb{N}$. A close form expression for F^{*} is equal to $\lim_{\xi\to\infty}F^{(\xi)}{=}F^{*}{=}S^{*}{=}\mathrm{vec}^{-1}((I-\mathbb{S})^{-1}\mathrm{vec}(I))$. Here, vec is an operator that stacks the columns of a matrix one after the next into a column vector. This above closed form equation and the proof of the convergence of Eq.(5) can be found in (Yang, Prasad, and Latecki 2013). We can acquire that the iterative approach on the original graph N defined by Eq.(5) yields the same similarities as the tensor product graph diffusion process on $\mathbb{N}$ for a sufficient number q of iterations(Shu and Latecki 2015; 2016; Yang, Prasad, and Latecki 2013). For attribute a_m, after q is reached, we can get a diffusion matrix $H(a_m)$ (that is, $F^{(q+1)}$) which is also a new affinity matrix related to all values of attributes a_m and a_u. In addition, in first $f(a_m)$ rows of $H(a_m)$, each row respectively correspond to each value's embedding vector of attribute a_m.

Next, we choose the tensor product graph diffusion in TAVE rather than the general graph diffusion since the former can discover higher order relations compared to the latter. The diffusion process also can propagate the intra- and inter-attribute value similarity information on the tensor product graph and make each value's embedding vector of attribute a_m, which is derived from the diffusion matrix $H(a_m)$, contain these two kinds of information. Hence, these information can be regarded as each value's intrinsic features and used to quantify the differences between attribute values.

Finally, We further analyze the time complexity of Algorithm 1. In Algorithm 1, the computational costs of four steps are severally $O(tn{+}tnk{+}n^2{+}nk)$, $O(t)$, $O(t^2)$ and $O(\sum_{m=1}^{t}p_m^3)$ where p_m is the sum of the number of all attribute values from a_m and a_u. Therefore, the time cost of Algorithm 1 is $O(tn{+}tnk{+}n^2{+}nk{+}t{+}t^2{+}\sum_{m=1}^{t}p_m^3)$. Since from the practical view point, k usually is a constant, the time complexity of TAVE is $O(tn{+}n^2{+}t^2{+}\sum_{m=1}^{t}p_m^3)$ and it is lower than CNS.

Experiments

In this section, we test our method on several data sets and compare it with a series of baselines in order to validate the usefulness of the proposed method.

Data sets

Ten UCI data sets were used in experiments. For the minority of them, we did the discretization algorithm on numerical attributes so as to change them into nominal ones. These ten data sets were as follows: Teaching Assistant Evaluation (Tae for short), Solar Flare Data1 (Solar Data1), Liver Disorders (Liver), MONK's Problems (Monks), Balance Scale (Balance), Tic-Tac-Toe Endgame (Tic), German Credit Data (German), Contraceptive Method Choice (CMC), Chess(King-Rook vs. King-Pawn) (Chess(KRvsKP)), and Chess (King-Rook vs. King) (Chess(KRvsK)). Table 2 lists their main characteristics and No. represents the serial number of each data set. Note that all attributes are totally independent in Monks and Balance data sets here.

Table 2: Data sets used in experiments

No.	Data sets	Objects	Attributes	Classes
#1	Tae	151	5	3
#2	Solar Data1	323	10	3
#3	Liver	345	6	2
#4	Monks	432	6	2
#5	Balance	625	4	3
#6	Tic	958	9	2
#7	German	1000	20	2
#8	CMC	1473	9	3
#9	Chess(KRvsKP)	3196	36	2
#10	Chess(KRvsK)	28056	6	18

Experimental Settings

The first setting is in relation to the baseline methods. In the light of two chief kinds of methods for nominal data clustering, we used HAM, HDM, CNS and ADM as the baseline methods where the dissimilarity degrees between nominal objects were regarded as the distance metrics, and also selected DVE, TVE and CDE as the baseline methods, by which the Euclidean distance can be applied to the embedded nominal data set. Here, TVE and DVE were severally based on TF-IDF and the generation of dummy attributes. To make TF-IDF available for general nominal data sets, in this paper, we thought of each nominal attribute as an unique document collected in a corpus and considered each different value in an attribute as a ”term”, and then TF-IDF was severally applied to each attribute. In addition, there are two significant features which distinguish TAVE from other approaches mentioned in this paper: obtaining the valuable information from each nominal object and its k nearest neighbors and propagating the similarities information by the tensor product graph diffusion process. Accordingly, we designed a new baseline method called AVE to verify that these two different features were capable of powerfully exerting influence on the clustering performance. In AVE, each attribute value's weight $\widetilde{a}_m^{\beta}$ is calculated only by each value occurrence frequency within attribute a_m of n objects, and for each attribute a_m, the diffusion matrix $H(a_m)$ is constructed only by concatenating the corresponding intra-attribute value similarity matrix $M(a_m)$ and inter-attribute value similarity matrix $Q(a_m)$. Actually, AVE is roughly equivalent to the proposed TAVE where the tensor product graph diffusion process is not implemented and the parameter k=0 in Eq.(2).

The second setting is in regards to the clustering methods. The main goal of this paper does not concentrate on intending the high-powered clustering algorithms, and TAVE is quite distinct from HAM, HDM, CNS and ADM . Hence, two simple clustering methods K-means and K-modes are applied for experiments. K-means is the prevalently used numerical clustering algorithm, and K-modes originating from the homogeneous K-means is the most popular nominal clustering approach. We conducted nine strategies for clustering on each data set: K-modes with HAM, K-modes with HDM, K-modes with CNS, K-modes with ADM, K-means with DVE, K-means with TVE, K-means with CDE, K-means with AVE and K-means with the proposed TAVE.

The next setting is in concern of the evaluation criterion. To establish a fair comparison between all methods, the cluster number was fixed to the number of classes in each data set and two commonly used evaluation criteria for clustering were taken here, namely, F-score and the standard normalized mutual information (NMI)(Manning, Raghavan, and Schtze 2008). F-score and NMI were averaged on 100 independent runs for each data set. The larger values of F-score and NMI indicate the better clustering performance.

The last setting is related to the selection of parameters. In all baseline methods, the optimal parameters were employed for each data set. In TAVE, we uniformly set the maximum iteration number q=20 for each data set because TAVE is insensitive to the parameter q. Furthermore, in TAVE, the choice of the parameter k[1] was based on the number n of objects and relied on the following Eq.(6) for each data set in all experiments.

$$k = \begin{cases} 10 & n < 1000 \\ 100 & 1000 \leq n < 10000 \\ 1000 & n \geq 10000 \end{cases} \tag{6}$$

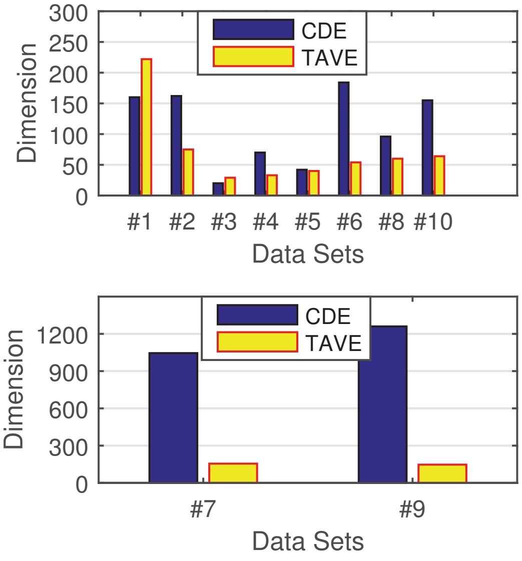

Figure 2: Comparison of dimensions on ten UCI data sets

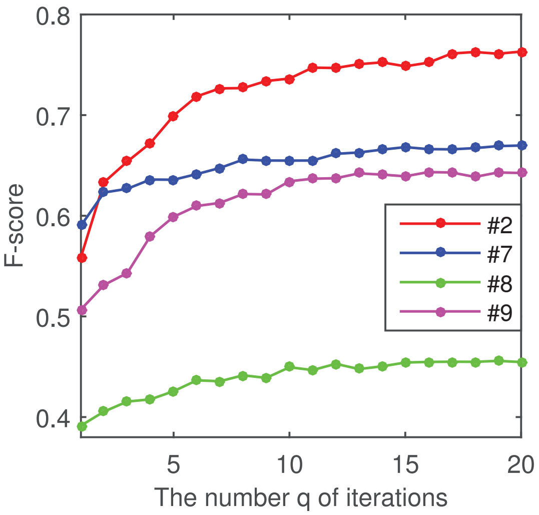

Figure 3: Sensitivity test of the parameter q on four data sets

Results Analysis

Experimental results of F-score and NMI on ten UCI data sets are individually reported in Tabel 3 and 4 and the best clustering performance is marked in bold face. The evaluation of dimensions between CDE and TAVE are shown in Figure 2 and each method is represented with a different color level. The sensitivity of the parameter q is examined in Figure 3 and the different color level symbolizes distinct data set.

Comparison with HAM, HDM ,CNS and ADM As shown in Table 3 and 4, CNS and ADM are a little better than the homogeneous method HAM for measuring cluster-

[1] Due to space limitations, for each data set, we do not show the detailed results based on different k here.

Table 3: Comparison of F-score on ten UCI data sets

Data	K-modes				K-means				
sets	HAM	HDM	CNS	ADM	DVE	TVE	CDE	AVE	TAVE
#1	0.4217	0.4067	0.4047	0.3691	0.4113	0.3742	0.3856	0.3587	**0.4226**
#2	0.5070	0.5506	0.6215	0.5463	0.5079	0.6843	0.5314	0.6528	**0.7631**
#3	0.5514	0.5487	0.5621	0.5546	0.5457	0.5625	0.5468	0.5419	**0.5745**
#4	0.5244	0.5085	—	—	0.5222	—	0.5211	—	**0.5263**
#5	0.4260	0.4284	—	—	0.4195	—	0.4328	0.4210	**0.4478**
#6	0.5379	0.5313	0.5266	0.5288	0.5379	**0.6060**	0.5368	0.5822	0.5420
#7	0.5607	0.5662	0.6480	0.5768	0.5448	0.6132	0.5795	0.6011	**0.6698**
#8	0.3870	0.3952	0.3972	0.3912	0.3667	0.4168	0.3745	0.3848	**0.4547**
#9	0.5354	0.5336	0.5867	0.5409	0.5083	0.5394	0.5129	0.5322	**0.6425**
#10	0.1040	0.1206	0.1228	0.1247	0.1223	0.1250	0.1211	0.1190	**0.1270**

Table 4: Comparison of NMI on ten UCI data sets

Data	K-modes				K-means				
sets	HAM	HDM	CNS	ADM	DVE	TVE	CDE	AVE	TAVE
#1	0.0461	0.0508	0.0323	0.0395	0.0393	0.0450	0.0475	0.0234	**0.0756**
#2	0.0236	0.0213	0.0300	0.0239	0.0231	0.0286	0.0332	0.0260	**0.0377**
#3	0.0198	0.0185	0.0159	0.0229	0.0190	0.0250	**0.0257**	0.0186	0.0231
#4	0.0200	0.0138	—	—	0.0272	—	0.0307	—	**0.0335**
#5	0.0303	0.0324	—	—	0.0363	—	0.0612	0.0392	**0.0711**
#6	0.0137	0.0146	0.0070	0.0100	0.0097	0.0018	0.0095	0.0020	**0.0155**
#7	0.0090	0.0105	0.0072	0.0089	0.0080	0.0069	0.0118	**0.0246**	0.0065
#8	0.0270	0.0291	**0.0375**	0.0362	0.0320	0.0309	0.0281	0.0311	0.0194
#9	0.0108	0.0089	0.0092	0.0092	0.0046	0.0120	0.0072	0.0056	**0.0184**
#10	0.0652	0.1082	0.1288	0.1287	0.1188	0.1168	0.1152	0.1219	**0.1303**

ing quality, but they can not work on Monks and Balance data sets, in which the attributes are totally independent of each other. Except for NMI only on one data set, TAVE outperform HAM, HDM, CNS and ADM on almost all the evaluation measures. Moreover, we can see from Table 3 and 4 that this class of methods based on the specific distance metrics also underperforms the clustering performance of other attribute value embedding representation methods, such as TVE and CDE.

Comparison with DVE and TVE As Table 3 and 4 indicates, the performance of DVE and TVE respectively are inferior to and as good as other homogeneous methods (e.g., CDE and AVE), and when they are taken for a comparison with TAVE, except for F-score only on one data set, TAVE is all superior to DVE and TVE. Furthermore, our TAVE is competent for data sets with totally independent attributes in comparison with TVE.

Comparison with CDE When the proposed TAVE is compared with CDE, although table 4 presents that TAVE obtains the disappointing NMI on 3 out 10 data sets, TAVE takes on a very satisfying F-score on all data sets in Table 3. In addition, Figure 2 demonstrates that TAVE transform a nominal object into a numerical object with fewer dimensions than CDE for all data sets except two data sets. Specially, the dimensions of an object generated by CDE are almost eight times as many as an object's dimensions produced by TAVE on German and Chess(KRvsKP) data sets.

Comparison with AVE The comparison between AVE and our TAVE aims to further test the influence of k nearest neighbors and the tensor product graph diffusion process. Table 3 shows that TAVE can achieve the better F-score than AVE on all data sets except Tic data set, and Table 4 describe that TAVE can perform better than AVE for all data sets except German and CMC data sets. In general , experimental results reflect the fact that more useful information is captured by utilizing each object's k nearest neighbors and tensor product diffusion process in TAVE.

Parameter Sensitivity Experiments on four UCI data sets were conducted to see whether F-score of TAVE is sensitive to the number q of diffusion iterations. We can see from Figure 3 that when $q{>}15$, F-score for four data sets can achieve a very stable results. Therefore, we set $q{=}20$ for all experiments. Here, we do not show F-score on other six data sets because the results on each data set do not present the very obvious differences for different q.

Conclusions

This paper proposes an effective tensor product graph diffusion related attribute value embedding method TAVE which can make the Euclidean distance usable for nominal data clustering. Compared to other baseline methods for nominal data clustering, our method (1) enables to show its universality because it is built on the foundation of information theories and graph diffusion, (2) demonstrates the better clustering performance. There is, however, no free lunch. The proposed TAVE have a little higher time complexity because it should search k nearest neighbors for each object on the whole data set. In the future work, we will adopt a new method to quickly find k nearest neighbors of each object.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No.:61302157.

References

Agresti, A. 2007. *An introduction to categorical data analysis*. Wiley-Blackwell.

Ahmad, A., and Dey, L. 2007. A method to compute distance between two categorical values of same attribute in unsupervised learning for categorical data set. *Pattern Recognition Letters* 28(1):110 – 118.

Berry, M. W., ed. 2003. *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer, New York.

Bock, H. H. 2000. *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Boriah, S.; Chandola, V.; and Kumar, V. 2008. Similarity measures for categorical data: A comparative evaluation. In *SDM*, 243–254. SIAM.

Cai, D.; He, X.; and Han, J. 2005. Document clustering using locality preserving indexing. *IEEE Trans. Knowl. Data Eng.* 17(12):1624–1637.

dos Santos, T. R., and Zrate, L. E. 2015. Categorical data clustering: What similarity measure to recommend? *Expert Systems with Applications* 42(3):1247 – 1260.

Huang, Z. 1998. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery* 2(3):283–304.

Jia, H.; m. Cheung, Y.; and Liu, J. 2016. A new distance metric for unsupervised learning of categorical data. *IEEE Transactions on Neural Networks and Learning Systems* 27(5):1065–1079.

Jiang, X.; Liu, W.; Cao, L.; and Long, G. 2015. Coupled collaborative filtering for context-aware recommendation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, 4172–4173. AAAI Press.

Le, S. Q., and Ho, T. B. 2005. An association-based dissimilarity measure for categorical data. *Pattern Recognition Letters* 26(16):2549 – 2557.

Li, F.; Xu, G.; and Cao, L. 2015. *Coupled Matrix Factorization Within Non-IID Context*. Cham: Springer International Publishing. 707–719.

Liu, C., and Cao, L. 2015. *A Coupled k-Nearest Neighbor Algorithm for Multi-label Classification*. Cham: Springer International Publishing. 176–187.

Manning, C. D.; Raghavan, P.; and Schtze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Pang, G.; Cao, L.; and Chen, L. 2016. Outlier detection in complex categorical data by modeling the feature value couplings. In Kambhampati, S., ed., *IJCAI*, 1902–1908. IJCAI/AAAI Press.

Shu, L., and Latecki, L. J. 2015. Transductive domain adaptation with affinity learning. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, 1903–1906. New York, NY, USA: ACM.

Shu, L., and Latecki, L. J. 2016. Integration of single-view graphs with diffusion of tensor product graphs for multi-view spectral clustering. In Holmes, G., and Liu, T.-Y., eds., *Asian Conference on Machine Learning*, volume 45 of *Proceedings of Machine Learning Research*, 362–377. Hong Kong: PMLR.

Songlei Jian, Longbing Cao, G. P. K. L. H. G. 2017. Embedding-based representation of categorical data by hierarchical value coupling learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 1937–1943.

Soong, T. T. 2004. *Fundamentals of Probability and Statistics for Engineers*. Chichester; Hoboken, NJ: Wiley.

Suits, D. B. 1957. Use of dummy variables in regression equations. *Journal of the American Statistical Association* 52(280):548–551.

Wang, C.; Cao, L.; Wang, M.; Li, J.; Wei, W.; and Ou, Y. 2011. Coupled nominal similarity in unsupervised learning. In Macdonald, C.; Ounis, I.; and Ruthven, I., eds., *CIKM*, 973–978. ACM.

Wang, C.; Dong, X.; Zhou, F.; Cao, L.; and Chi, C. H. 2015. Coupled attribute similarity learning on categorical data. *IEEE Transactions on Neural Networks and Learning Systems* 26(4):781–797.

Yang, X.; Prasad, L.; and Latecki, L. J. 2013. Affinity learning with diffusion on tensor product graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(1):28–38.

Zdravevski, E.; Lameski, P.; Kulakov, A.; and Kalajdziski, S. 2015. Transformation of nominal features into numeric in supervised multi-class problems based on the weight of evidence parameter. In *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 169–179.