Expert Systems With Applications 150 (2020) 113206

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Non-numerical nearest neighbor classifiers with value-object hierarchical embedding

Sheng Luo[a], Duoqian Miao[b,c,*], Zhifei Zhang[b,c,*], Zhihua Wei[b,c]

[a] School of Computer and Information, Shanghai Second Polytechnic University, Shanghai 201209, China
[b] Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
[c] Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China

ARTICLE INFO

Article history:
Received 3 September 2018
Revised 13 January 2020
Accepted 13 January 2020
Available online 15 January 2020

Keywords:
Non-numerical classification
Categorical data
Nearest neighbor classifier
Data complexity
Attribute reduction

ABSTRACT

Non-numerical classification plays an essential role in many real-world applications such as DNA analysis, recommendation systems and expert systems. The nearest neighbor classifier is one of the most popular and flexible models for performing classification tasks in these applications. However, due to the complexity of non-numerical data, existing nearest neighbor classifiers that use the overlap measure and its variants cannot capture the inherent ordered relationship and statistic information of non-numerical data. This phenomenon leads to the classification limitation of nearest neighbor classifiers in non-numerical data environments. To overcome this challenge, we propose a novel object distance metric, i.e., value-object hierarchical metric (VOHM), which is able to capture inherent ordered relationships within non-numerical data. Then, we construct two nearest neighbor classifiers, i.e., the value-object hierarchical embedded nearest neighbor classifier (VO-*k*NN) and the two-stage value-object hierarchical embedded nearest neighbor classifier (TSVO-*k*NN), which take advantages of both VOHM and non-numerical feature selection. Experiments show that both VO-*k*NN and TSVO-*k*NN could mine more knowledge from data and achieve better performance than state-of-the-art classifiers in non-numerical data environments.

1. Introduction

The classification problem (Murphy, 2012) is a fundamental issue in the areas of machine learning, data mining, artificial intelligence, and expert systems etc. It plays an essential role in many applications, including sentiment analysis, spam filtering, image analysis, text analysis, and DNA sequence analysis, etc. There exist a lot of methods to solve classification tasks such as logistic regression (LR) (Murphy, 2012; Walker & Duncan, 1967), random forest(RF) (Breiman, 2001; Ho, 1995), support vector machines(SVMs) (Catanzaro, Sundaram, & Keutzer, 2008; Cortes & Vapnik, 1995), artificial neural networks(ANN) and deep learning(DL) (Bengio, Courville, & Vincent, 2013; Lecun, Bengio, & Hinton, 2015) etc., which have achieved great success in numerical environments. However, it is a big challenge to apply these models to solve classification tasks in non-numerical environments due to the lack of useful non-numerical distance metrics for evaluating the relations between objects.

Non-numerical (or categorical) data is a widely used data type in expert systems. For example, Table 1 is a staff table instance with non-numerical values. It is convenient for people to acquire knowledge from non-numerical values. However, most existing machine learning algorithms are difficult to deal with the same problem as humans do, because, in the perspective of machine learning algorithms, non-numerical data does not contain the same semantics or context that humans can easily capture and understand. Hence, narrowing the gap between algorithms and humans has become one of the critical factors to improve the classification performance of the algorithms in non-numerical data environments. To solve the non-numerical classification problem, the most commonly used strategy is to convert non-numerical data into numerical data and process it with machine learning algorithms (Buttrey, 1998) such as nearest neighbor classifiers (Chen & Guo, 2015; Hu, Yu, & Xie, 2008; Liu, Cao, & Yu, 2014). The *k*-nearest neighbor classifier (*k*NN) is one of the most frequently used non-parametric classification models in expert systems (Mller et al., 2019; Rodger, 2014), because the model has no training phase and easy implementation. Therefore, the problem of how to convert non-numerical data into numerical values becomes the bottleneck of the *k*NN classifiers for performing non-numerical classification tasks.

* Corresponding authors at: Department of Computer Science and Technology, Tongji University, Shanghai 201804, China.
E-mail addresses: tjluosheng@gmail.com (S. Luo), dqmiao@tongji.edu.cn (D. Miao), zhifeizhang@tongji.edu.cn (Z. Zhang), zhihua_wei@tongji.edu.cn (Z. Wei).

https://doi.org/10.1016/j.eswa.2020.113206
0957-4174/

Table 1
An instance of the staff database.

Name	Majors	Educations	Hobbies	Classes
Abel	Math	Doctor	Dancing	C_1
Chad	Math	Master	Singing	C_1
Carter	Economics	Bachelor	Dancing	C_1
Duncan	Logic	Bachelor	Singing	C_2
Frank	Philosophy	Bachelor	Singing	C_2
Jessie	Economics	Bachelor	Swimming	C_2

The overlap metric (or simply match coefficients) (Boriah, Chandola, & Kumar, 2008) is a commonly used discrepancy measure which is able to convert data from non-numerical form into numerical form. However, due to the neglect of the different contributions of attributes in non-numerical classification problems, the overlap metric can not meet the requirement of real-world applications. Although the attribute weighting (Chen & Guo, 2015; Chen, Ye, Guo, & Zhu, 2016; Morlini & Zani, 2012) based approach satisfies these requirements, it still fails to capture latent ordered relationships that exist in non-numerical values. For example, in Table 1, according to common sense, the distance of values between 'Bachelor' and 'Doctor' is obviously greater than that between 'Bachelor' and 'Master'. However, capturing this latent ordered information is beyond the scope of the overlap measure and attribute weighting based measures. Therefore, when the classification algorithm processes non-numerical data, how to mine latent ordered information from non-numerical values becomes one of the basic problems to be solved. Furthermore, there also exists another complicated dependency relationship between non-numerical values from different attributes. For example, the association rule 'Beer' $\Rightarrow$ 'Diapers' found in the sales data would indicate that if a customer buys beer, it is likely also to buy diapers with a certain probability (Agrawal & Srikant, 1994; Han, Pei, & Yin, 2000; Zaki, 2000). In other words, It means that there exist dependency relationships between non-numerical values from different attributes. Hence, another question is arose, namely, how to represent and capture the second relationship in non-numerical data environments. In summary, these challenges put new requirements to update the distance measure for non-numerical data. Data science (Cao, 2017a; 2017b) shows that there exist complicated relationships in data, and a lot of knowledge required intelligence algorithm to discover.

To overcome these problems, in this paper, we construct a novel distance metric, called value-object hierarchical metric (VOHM), to update the strategy of distance measure for non-numerical data. VOHM could learn the latent ordered relationship from the value-object hierarchy structure. At the value level, the VOHM handles all values in a probability perspective, which could capture the latent ordered relationship of values relative to the class label distribution. And, at the object level, VOHM treat the distance of each object pair as the total sum of the discrepancies from the value level. Then, we proposed a nearest neighbor classifier that takes advantages of both VOHM and attribute reduction. Firstly, to avoid the curse of dimensionality and reduce calculation, we employ the rough set theory based attribute reduction strategy to select the required attributes. Then, we equip up the nearest neighbor classifier with VOHM to perform the classification task for the selected attribute filtered data set. More specifically, we developed two nearest neighbor classifiers, i.e., the value-object hierarchical embedded nearest neighbor classifier (VO-kNN) and the two-stage value-object hierarchical embedded nearest neighbor classifier (TSVO-kNN).

The main contributes of this paper are summarized as follows:

- A new distance measure for non-numerical data, i.e., VOHM, is proposed. VOHM enables us to capture the latent ordered relationship, which gives more knowledge than the overlap measure and the weighted overlap measure.
- We propose a value-object hierarchical embedded nearest neighbor classifier that takes advantage of VOHM.
- We propose a two-stage value-object hierarchical embedded nearest neighbor classifier to perform the classification task on the reduced data set to avoid the curse of dimensionality.

The rest of the paper is organized as follows. In Section 2, we briefly do an overview of existing classifiers for non-numerical data. In Section 3, we formulate the research problem and introduce the preliminary notions. Section 4 defines the VO-kNN classifier. In Section 5, we design a feature section algorithm based on rough set theory. In Section 6, we conduct experiments to show the advantage of our model and algorithm. Finally, we conclude this work in Section 7.

2. Overview of existing classifiers for categorical data and related work

In the literature, researchers have proposed a lot of classifiers for categorical data. This section will present an overview of them as follows.

2.1. Decision tree based categorical classifier

2.1.1. Iterative dichotomiser 3 (ID3)

ID3 is a classifier invented by Quinlan (1986), which uses information gain to generate a decision tree from the training data set and predicts the class label of a new sample according to the trained tree model. Given a dataset $\mathcal{S}$, we let Z be a discrete random variable with possible values $\tilde{Z} \triangleq \{z_1, z_2, \ldots, z_r\}$ representing an attribute. We also let L be a discrete random variable with possible values $\tilde{L} \triangleq \{l_1, l_2, \ldots, l_t\}$ representing the decision attribute. Then, the information gain (IG) of an attribute Z is a measure of the entropy discrepancy from before to after the set $\mathcal{S}$ is split on the attribute Z, i.e.,

$$IG(\mathcal{S}, Z) = H(L) - H(L|Z), \tag{1}$$

where $H(L)$ is the entropy of the decision attribute L, i.e.,

$$H(L) = -\sum_{l \in \tilde{L}} p(l) \cdot \log_2 p(l), \tag{2}$$

where $p(l) \triangleq Pr(L = l)$ is the ratio of the number of elements with decision l to the number of elements in the dataset $\mathcal{S}$, and $H(L|Z)$ is the conditional entropy of L given attribute Z, i.e.,

$$\begin{aligned} H(L|Z) &= \sum_{z \in \tilde{Z}} p(z) \cdot H(L|z) \\ &= -\sum_{z \in \tilde{Z}} p(z) \cdot \sum_{l \in \tilde{L}} p(l|z) \cdot \log_2 p(l|z), \end{aligned} \tag{3}$$

where $p(l|z) \triangleq Pr(L = l|Z = z)$ is the ratio of the number of elements with the decision l to the number of elements that satisfy the condition $Z = z$ in the dataset $\mathcal{S}$.

The ID3 algorithm firstly regards the training data set as the root of the tree. Then it will calculate all information gains of unused attributes and selects one attribute which has maximum information gain to split the set. According to this rule, the algorithm constructs a decision tree iteratively. After the decision tree is established, the class label of a new object output by the tree model is the value of the node that the object arrives. In the context of the ID3 algorithm, the terminal node represents the class label to which an object belongs.

2.1.2. C4.5

The disadvantage of ID3 is that the information gain tends to choose the attribute with more values. However, in some cases, these attributes may not provide much valuable information. To avoid this drawback, C4.5 (Quinlan, 1993) uses the information gain rate to select the split attribute from the candidate attributes. The information gain rate of an attribute Z is defined as follows:

$$GainRatio(\mathcal{S}, Z) = \frac{IG(\mathcal{S}, Z)}{SplitInfo_Z(\mathcal{S})}, \tag{4}$$

where $SplitInfo_Z(\mathcal{S})$ represents the split information which is defined as follows:

$$SplitInfo_Z(\mathcal{S}) = -\sum_{j=1}^{N} \frac{|\mathcal{S}_j|}{|\mathcal{S}|} \log_2 \frac{|\mathcal{S}_j|}{|\mathcal{S}|}, \tag{5}$$

where $|\cdot|$ is a function used to calculate the cardinality of a set. $\mathcal{S}_j (j = 1, 2, \ldots, N)$ is a subset of the set $\mathcal{S}$, and its elements are determined by the values of the selected attribute Z. The tree model construction process of C4.5 and ID3 is similar, but the different is that the strategy for selecting candidate attributes in C4.5 is the information gain rate.

2.1.3. Classification and Regression Trees (CART)

Similar to ID3 and C4.5, CART (Breiman, Friedman, Olshen, & Stone, 1984; Loh, 2012) is another decision tree classifier for categorical data.The difference between CART and ID3, C4.5 is that the former uses attributed split Gini impurity (ASGI) to select an attribute to split training data set and construct a decision tree model. ASGI is defined as follows:

$$ASGI(\mathcal{S}, Z) = \sum_{i=1}^{l} \frac{N_i}{N} Gini(\mathcal{S}_i), \tag{6}$$

where $\mathcal{S}_i (i = 1, 2, \ldots, l)$ is a subset of $\mathcal{S}$ and the elements of $\mathcal{S}_i$ are determined by splitting $\mathcal{S}$ using the values of the attribute Z. N_i and N are cardinalities of $\mathcal{S}_i$ and $\mathcal{S}$, respectively. $Gini(\mathcal{S}_i)$ is the Gini impurity of $\mathcal{S}_i$, which is defined as,

$$Gini(\mathcal{S}_i) = \sum_{k=1}^{J} q_k(1 - q_k) = 1 - \sum_{k=1}^{J} q_k^2, \tag{7}$$

where the items of $\mathcal{S}_i$ have $J (J \le N_i)$ classes, and $q_k (k = 1, 2, \ldots, J)$ is the fraction of items labeled with class k in the set $\mathcal{S}_i$.

In addition to ID3, C4.5 and CART, there are also some classification models based on the decision tree for categorical data such as chi-squared automatic interaction detector (CHAID) (Kass, 1980), quick unbiased and efficient statistical tree (QUEST) (Wei-Yin, 2014), classification rule with unbiased interaction selection and estimation (CRUISE) (Kim & Loh, 2001), generalized unbiased interaction detection and estimation (GUIDE) (Loh, 2009), conditional inference trees (CTREE) (Hothorn, Hornik, & Zeileis, 2006), and various modifications of decision tree algorithms (Kim, 2016; Zhao & Li, 2017).

2.2. Naive Bayes classifier

The naive Bayes (NB) classifier (Hand & Yu, 2001; Murphy, 2012) is a family of simple probabilistic classifiers, which holds the assumption of attribute conditional independence. Given a train set $\mathcal{T} = \{\boldsymbol{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$ and the assumption, we have the definition of naive Bayes models, i.e.,

$$p(\boldsymbol{x}^{(i)}|y^{(i)} = c, \boldsymbol{\Theta}) = \prod_{j=1}^{m} p(x_j^{(i)}|y^{(i)} = c, \Theta_{jc}), \tag{8}$$

where m is the number of features, and $\boldsymbol{\Theta}$ is the parameter of the naive Bayes classifier which has nm parameters for n classes and m features, and Θ_{jc} is the parameter of the conditional probability distribution which represents the distribution of the jth feature given the condition $y^{(i)} = c$. Unlike the classifiers based on decision trees, the naive Bayes model is a parametric model, which means it needs data to fit. By using maximum likelihood estimator (MLE), the goal of training NB classifiers is to maximize the following log-likelihood, i.e.,

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \log \prod_{i=1}^{N} p(y^{(i)}) p(\boldsymbol{x}^{(i)}|y^{(i)}, \boldsymbol{\Theta}). \tag{9}$$

After training the model, the NB classifier uses $p(y|\boldsymbol{x}, \hat{\boldsymbol{\Theta}})$ to predict the class label for a new sample $\boldsymbol{x}$.

Due to the strict condition of the attribute independence hypothesis, Kononenko et al. relaxed this constraint and proposed semi-naive Bayes classifiers (SNBC) (Kononenko, 1991). According to the criteria of attribute dependence, a lot of semi-naive Bayes classifier are proposed such as tree augmented naive Bayes (TAN) (Jiang, Cai, Wang, & Zhang, 2012a; Zheng & Webb, 2011), weighted averaged one-dependent estimator (WAODE) (Jiang, Zhang, Cai, & Wang, 2012b), etc.

2.3. Instance-based learning

A classifier based on instance-based learning (Daelemans & Van den Bosch, 2005; Russell & Norvig, 2016) is another nonparametric classification model that can be applied to the classification problem for categorical data. One of the simplest instance-based classifiers is the k-nearest neighbor (kNN) classifier. The bottleneck of this approach is the problem of how to evaluate the relationship between objects. For the numerical data environments, there exist a lot of distance measures for objects such as the Euclidean distance, the Manhattan distance, the Chebyshev distance, etc. However, in the non-numerical data environment, these distance measure mentioned above are not easy to apply to measure categorical data distance. The simplest solution to handle non-numerical data is to convert it into binaries, and then it could be treated like numerical data in kNN classification models (Buttrey, 1998). For example, a commonly used convert strategy in kNN classifier for categorical data is overlap measure (Boriah et al., 2008) also known as the Hamming distance, which is defined as,

$$\delta^{(ol)}(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \sum_{k=1}^{d} \mathbb{I}(x_k^{(i)}, x_k^{(j)}), \tag{10}$$

where $\boldsymbol{x}^{(i)}$ and $\boldsymbol{x}^{(j)}$ are feature vectors with the dimension d, and $\mathbb{I}(u, v)$ is an indicator function which means that if $u = v$ is satisfied then $\mathbb{I}(u, v) = 1$ otherwise $\mathbb{I}(u, v) = 0$. The disadvantage of this approach is obvious, that is, the conversion will lose the original relationship within data, and the converted data will be limited to a hypothetical ordered relationship. Another bottleneck of this distance measure is the assumption that all attributes play an equal role for classifying non-numerical objects. However, in real applications, it is always true that not all attributes are valuable for performing classification tasks and for predicting the label distribution of a sample. For example, noise attributes (Li, Wen, Yu, & Zhou, 2013) do not contribute to predicting label in the high feature dimensional environment. A natural solution is to perform feature selection (Hu et al., 2008) in advance. Similarly, the classifier can give different weights to each object (Jahromi, Parvinnia, & John, 2009) or attribute (Chen & Guo, 2015; Morlini & Zani, 2012) or distance (Alamelu, Milind, & Santhosh, 2013; Jiang, Cai, Wang, & Zhang, 2014) to distinguish its contributions. Although the existing methods improved the classification ability of kNN classifiers from different angles, however, it still cannot handle non-numerical classification problems well to achieve the performance

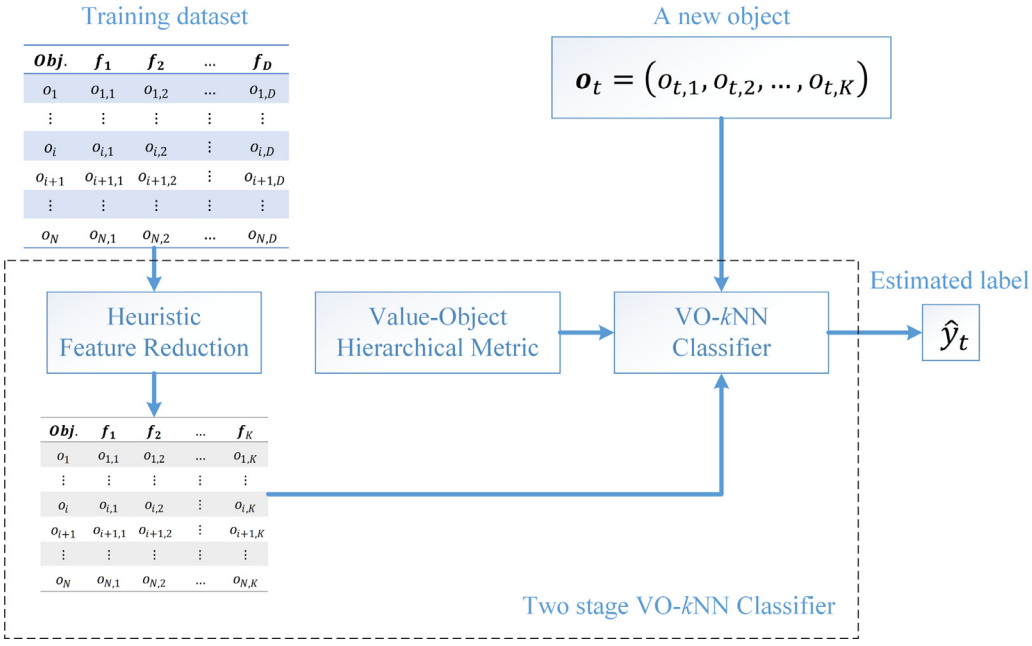


Fig. 1. The architecture of VO-*k*NN classifier and TSVO-*k*NN classifier.

similar to that of dealing with numerical data. The key to the problem is the distance measure of non-numerical objects should have the same ability as the distance measure for numerical objects. Boriah et al. (2008) conducted a comparative study of some commonly used categorical metrics. Although these metrics can be embedded in the *k*NN classifier, they ignore the statistics between the categorical data and the value-object hierarchical structure, which affects the performance of the modified *k*NN classifier. Besides, there also exist some factors that affect the classification accuracy of the model, such as the neighbor size k (Gou et al., 2019). Therefore, in the non-numerical data environment, these factors should be paid more attention to decrease the impact on classification performance.

2.4. Other methods

Based on kernel density estimation, Chen et al. proposed a linear classifier for categorical data (Chen et al., 2016). In addition to this, there are a lot of methods (Garc et al., 2009; Liu et al., 2014) similar to mutual information for measuring the feature weights of the categorical attributes, which are embedded in the classifier for classifying categorical objects.

3. Problem formulation and framework

In what follows, the dataset $\mathcal{T}$ consists of data objects, i.e.,$\mathcal{T} \triangleq \{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(m)}, y^{(m)})\}$, where m is the size of the set $\mathcal{T}$, and $y^{(i)}$ is the class label of the object $\boldsymbol{x}^{(i)}$. Each object $\boldsymbol{x}^{(i)}$ consists of categorical feature values, i.e., $\boldsymbol{x}^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_d^{(i)}\}$, where d is the total number of features. The categorical features means that each feature has a value set with a limited cardinality. For example, the categorical feature "sex" maybe has two values {'male', 'female'}.

Given a training data set, the goal of the classification task is to construct a reasonable classifier, which can correctly predict the class label of a new object. The framework of our proposed model VO-*k*NN is described in Fig. 1. Firstly, we perform the dimension reduction by using rough set theory to avoid the curse of dimension and reduce the calculation complexity of the algorithm. Then, we constructed a value-object hierarchical metric for categorical data and embedded it into *k*NN classifier to predict the class label for a new object. The main notations in this paper are listed in Table 2.

Table 2
List of main notations.

Variable	Explanation
$\mathcal{D}$	A training dataset
A	An attribute set
A_k	The kth attribute
m	The total number of objects
d	The dimension of features
V	The value set of all attributes
V_k	The value set of the kth attribute ($k = 1, 2, \ldots, d$)
V_{ai}	The ith value in V_a (i=1,2,...,$\lvert V_a \rvert$)
$\lvert V_k \rvert$	The cardinality of the set V_k ($k = 1, 2, \ldots, d$)
X	An object set
$\boldsymbol{x}^{(i)}$	The feature vector of the ith object in X
$\boldsymbol{y}$	The class label vector of $\mathcal{D}$
$y^{(i)}$	The class label of the ith object
$\mathcal{F}_{ai}^{(c)}$	The number of times of the value V_{ai} appeared in the class c
$\mathcal{F}_{ai}$	The number of times of the value V_{ai} appeared in all classes
$\xi_{V_{ai}}^{(c)}$	The proportion of $\mathcal{F}_{ai}^{(c)}$ to $\mathcal{F}_{ai}$

4. VO-*k*NN classifier

In this section, we will introduce the components of the VO-*k*NN classifier.

4.1. Value-object hierarchical metric for categorical data

Definition 1. A four tuple, $\mathcal{D} \triangleq \langle X, \boldsymbol{y}, A, V \rangle$, is called a training data set, which satisfies the following assumptions,

(1) $X = \{\boldsymbol{x}^{(i)}\}_{i=1}^{m}$ is a non-empty set, which element represents an object in one domain, and m is the total number of object.
(2) $\boldsymbol{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(m)})$ is a vector, which element represents the class label of the i^{th} object.
(3) $A = \{A_k\}_{k=1}^{d}$ represents the set of attributes, and d is the total number of attributes.

(4) $V = \{V_k\}_{k=1}^{d}$ is a set of V_k, and V_k is the value set of the attribute A_k.

For the convenience of the later description, we defined auxiliary information functions as follows.

Definition 2 (auxiliary information functions, AIF)**.** Given a training data set $\mathcal{D} \triangleq \langle X, \boldsymbol{y}, A, V\rangle$, two auxiliary information functions are defined as follows: $\psi_k : 2^X \to V_k$, $(k = 1, \ldots, d)$, and the inverse function $\psi_k^{-1} : 2^{V_k} \to X$, $(k = 1, \ldots, d)$, that is,

$$\begin{aligned} &\psi_k(O) = W, \\ &\psi_k^{-1}(W) = O, \end{aligned} \tag{11}$$

where $O \subseteq X$, $W \subseteq V$.

From Table 1, for example, we can obtain the following equations: $\psi_1(\{'Abel', 'Chad'\}) = \{'Math'\}$ and $\psi_3^{-1}(\{'Dancing'\}) = \{'Abel', 'Carter'\}$.

Definition 3 (value level difference metric, VLDM)**.** Given a training dataset $\mathcal{D}$, we denote $\mathcal{F}_{ai}^{(c)}$ as the number of times that the value $V_{ai}(V_{ai} \in V_a, i = 1, 2, \ldots, |V_a|, a = 1, 2, \ldots, d)$ is classified into the class $c = \psi_d(\psi_a^{-1}(V_{ai}))$, and denote $\mathcal{F}_{ai}$ as the total number of times V_{ai} occurred in all classes, i.e.,

$$\mathcal{F}_{ai} = \sum_{c} \mathcal{F}_{ai}^{(c)}. \tag{12}$$

The frequency of value V_{ai} with label c is defined as,

$$\xi_{V_{ai}}^{(c)} = \mathcal{F}_{ai}^{(c)} / \mathcal{F}_{ai}. \tag{13}$$

Then, in the value level, the difference metric for value pair $\{(V_{ak}, V_{al})|\forall k, l < |V_a|\}$ in the ath feature $A_a(a = 1, 2, \ldots, d)$ is,

$$\delta^{(a)}(V_{ak}, V_{al}) = \sum_{c} \left\| \xi_{V_{ak}}^{(c)} - \xi_{V_{al}}^{(c)} \right\|^t, \quad t \in 2^{\{\mathbb{Z}^+ \cup 0\}}, \tag{14}$$

where $\|\cdot\|$ is a L^p-norm. In this work, if there is no special explanation about norms, then the norm we used is L^2-norm, i.e., $p = 2$.

The core idea of this definition is that we wish to tightly connect the values which occur with the same relative frequency in all classes.

Definition 4 (value-object hierarchical metric, VOHM)**.** The dissimilarity of the object pair $\{(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})|\forall i, j < m\}$ consists of the difference existing in all of the attributes value pairs. Therefore, the value-object hierarchical metric for the object pair $(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)})$ is defined as,

$$\Delta(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) = \sum_{a=1}^{d} \delta^{(a)}\left(\psi_a(\boldsymbol{x}^{(i)}), \psi_a(\boldsymbol{x}^{(j)})\right). \tag{15}$$

Unlike VDM (Stanfill & Waltz, 1986), MVDM (Cost & Salzberg, 1993), we do not need to consider the role of the features and objects because we filtered out the irrelevant features during the feature selection phase. Therefore, we ignore the weights of features and objects in the VLDM and VOHM definitions and treat the value-object as a whole from a hierarchical perspective.

Theorem 1. *Given $\forall \boldsymbol{b}, \boldsymbol{d}, \boldsymbol{q} \in X$, the value-object hierarchical metric $\Delta(\boldsymbol{b}, \boldsymbol{d})$ satisfies the following properties:*

1. $\Delta(\boldsymbol{b}, \boldsymbol{d}) \geq 0$
2. $\Delta(\boldsymbol{b}, \boldsymbol{d}) = \Delta(\boldsymbol{d}, \boldsymbol{b})$
3. $\Delta(\boldsymbol{b}, \boldsymbol{b}) = 0$
4. $\Delta(\boldsymbol{b}, \boldsymbol{d}) + \Delta(\boldsymbol{d}, \boldsymbol{q}) \geq \Delta(\boldsymbol{b}, \boldsymbol{q})$

Proof. (1) For any $t \in 2^{\{\mathbb{Z}^+\}}$, i.e., t is an even number, then the even power of any number is greater than or equal to 0, that is, $\|\cdot\|^t \geq 0$; For $t = 2^{\{0\}} = 1$, we have $|\cdot| \geq 0$. The sum of the non-negative values is non-negative, therefore, $\Delta(\boldsymbol{b}, \boldsymbol{d}) \geq 0$.

(2) Obviously, given $\forall a \in \{1, 2, \ldots, d\}$, the equation $\delta^{(a)}(\psi_a(\boldsymbol{d}), \psi_a(\boldsymbol{b})) = \delta^{(a)}(\psi_a(\boldsymbol{b}), \psi_a(\boldsymbol{d}))$ is always true. Then, we have the equation

$$\sum_a \delta^{(a)}(\psi_a(\boldsymbol{d}), \psi_a(\boldsymbol{b})) = \sum_a \delta^{(a)}(\psi_a(\boldsymbol{b}), \psi_a(\boldsymbol{d})),$$

i.e., $\Delta(\boldsymbol{d}, \boldsymbol{b}) = \Delta(\boldsymbol{b}, \boldsymbol{d})$.

(3) According to Eq. (14), if $d = b$, then $\delta^{(a)}(b, d) = 0$. Hence, we have the equation $\Delta(\boldsymbol{b}, \boldsymbol{d}) = 0$, when the equation $d = b(\forall d \in \boldsymbol{d}, b \in \boldsymbol{b})$ is satisfied.

(4) According to Eq. (15), we have,

$$\Delta(\boldsymbol{b}, \boldsymbol{d}) = \sum_a \sum_c \left\| \xi_{V_{ak}}^{(c)} - \xi_{V_{al}}^{(c)} \right\|^t$$

$$\Delta(\boldsymbol{d}, \boldsymbol{q}) = \sum_a \sum_c \left\| \xi_{V_{al}}^{(c)} - \xi_{V_{ap}}^{(c)} \right\|^t$$

$$\Delta(\boldsymbol{b}, \boldsymbol{q}) = \sum_a \sum_c \left\| \xi_{V_{ak}}^{(c)} - \xi_{V_{ap}}^{(c)} \right\|^t$$

Norms have the following properties: for matrices $\boldsymbol{A}, \boldsymbol{B}$, it satisfies

$$||\boldsymbol{A}|| + ||\boldsymbol{B}|| \geq ||\boldsymbol{A} + \boldsymbol{B}||$$

If we let $a' = \xi_{V_{ak}}^{(c)} - \xi_{V_{al}}^{(c)}$, $b' = \xi_{V_{al}}^{(c)} - \xi_{V_{ap}}^{(c)}$, and $c' = \xi_{V_{ak}}^{(c)} - \xi_{V_{ap}}^{(c)}$, then we have $||a'|| + ||b'|| \geq ||a' + b'|| = ||c'||$. It means that,

$$\sum_a \sum_j \left(\left\| \xi_{V_{ak}}^{(c)} - \xi_{V_{al}}^{(c)} \right\|^t + \left\| \xi_{V_{al}}^{(c)} - \xi_{V_{ap}}^{(c)} \right\|^t \right) \geq \sum_a \sum_j \left\| \xi_{V_{ak}}^{(c)} - \xi_{V_{ap}}^{(c)} \right\|^t$$

Hence, $\Delta(\boldsymbol{b}, \boldsymbol{d}) + \Delta(\boldsymbol{d}, \boldsymbol{q}) \geq \Delta(\boldsymbol{b}, \boldsymbol{q})$. □

Therefore, the value-object hierarchical metric can be used as a distance metric with the statistic information of attribute values. Obviously, this is an advantage that the overlap distance measure does not have. Actually, if we place some constraints on VOHM, then it becomes the overlap distance measure.

Theorem 2. *The overlap distance measure is a special case of the value-object hierarchical metric.*

Proof. If we place a constraint on VLDM, i.e.,

$$\delta^{(*a)}(V_{ak}, V_{al}) = \begin{cases} \mathbb{I}(\delta^{(a)}(V_{ak}, V_{al}) \geq 0), & V_{ak} \neq V_{al}; \\ 0, & Othervise. \end{cases} \tag{16}$$

where $\mathbb{I}(u)$ is a function which means if $u = ture$ is satisfied then $\mathbb{I}(u) = 1$ otherwise $\mathbb{I}(u) = 0$. Given $\forall \boldsymbol{b}, \boldsymbol{d} \in X$, then we have the special form $\Delta^*(\cdot, \cdot)$ of VOHM, i.e.,

$$\begin{aligned} \Delta^*(\boldsymbol{b}, \boldsymbol{d}) &= \sum_{a \in A} \delta^{(*a)}(\psi_a(\boldsymbol{b}), \psi_a(\boldsymbol{d})) \\ &= \sum_{a \in A} \mathbb{I}(\delta^{(a)}(\psi_a(\boldsymbol{b}), \psi_a(\boldsymbol{d})) \geq 0) \\ &= \sum_{a \in A} \delta^{(ol)}(\psi_a(\boldsymbol{b}), \psi_a(\boldsymbol{d})). \end{aligned} \tag{17}$$

Obviously, the third line of Eq. (17) is equal to the overlap measure, i.e. Eq. (10). □

4.2. VO-kNN classifier

In this work, we use VOHM to measure the distance of categorical objects and propose a value-object hierarchical metric embedded k nearest neighbor classifier (VO-kNN) to perform classification tasks. The following Algorithm 1 outlines the prediction algorithm of VO-kNN.

Algorithm 1 The prediction algorithm of VO-*k*NN for categorical data.

Input: $\mathcal{D} = \langle X, \boldsymbol{y}, A, V\rangle$: training dataset, k: the number of nearest neighbors, $\boldsymbol{x}'$: a query sample;
Output: y: the class label of $\boldsymbol{x}$.

```
//calculate the distance dist[] between x' and the element in X by using VOHM.
for i = 1, 2, ..., m do
    dist[i] ← Δ(x', x^(i)); //using Equation (15).
end for
sort(dist); //sort dist in ascending order.
C ← select(dist, k, X); select the k nearest neighbors.
return y ← arg max_c Σ_{(x^(i), y^(i)) ∈ C} 𝕀(c = y^(i)).
```

5. Two-stage VO-*k*NN classifier

Due to the feature correlation, we need to select some features that play a significant role in the classification task. In the numerical environment, there exist a lot of feature selection methods, for example, the principal component analysis (PCA). However, it is rare to find the feature selection method for non-numerical data except for rough set based methods.

The core issue of feature selection is to select a feature subset which has the same ability to distinguish all objects like the whole attributes. One of the commonly used metrics for evaluating the representation ability of the selected feature subset relative to the whole feature set is the rough approximation quality (Jia, Shang, Zhou, & Yao, 2016; Pawlak, 1998).

Definition 5. Given a training dataset $\mathcal{D} = \langle X, \boldsymbol{y}, A, V\rangle$, we let $\tilde{\boldsymbol{x}}^{(i)} = \langle \boldsymbol{x}^{(i)}, y^{(i)}\rangle$, $\tilde{X} = \{\tilde{\boldsymbol{x}}^{(i)}\}_{i=1}^{m}$, and denote C, D as the conditional attributes and the decision attribute, i.e., $C \cup D = A$. Then, given a feature subset $F \subseteq C$, the quality of rough approximation (QRA) (Pawlak, 1998) of $\tilde{X}/D$ w.r.t. F is defined as,

$$\gamma_F(D) = \frac{card(POS_F(D))}{card(\tilde{X})}, \tag{18}$$

where $card(\cdot)$ is a function to calculate the cardinality of a set. $\tilde{X}/D \triangleq \{X_1, X_2, \ldots, X_l\}$ is the set of equivalence class of $\tilde{X}$, which is divided by the values of the decision attribute D, and $POS_F(D)$ is the positive region (Pawlak, 1998) of the partition $\tilde{X}/D$ w.r.t. the equivalence relation induced by the attribute set F, which means that,

$$POS_F(D) = \bigcup\{[\boldsymbol{u}]_F | [\boldsymbol{u}]_F \subseteq X'\}, \tag{19}$$

where $\boldsymbol{u} \in X, X' \in \tilde{X}/D$ and $[\boldsymbol{u}]_F$ is the equivalence class of $\boldsymbol{u}$ that divided by the attribute set F.

Therefore, the feature selection problem became an optimization problem by using the QRA criterion. In other words, we want to search a feature subset F^* which has minimal cardinality and maximum QRA value, i.e.,

$$F^* = \arg\max_{F} \min_{|F|} \gamma_F(D). \tag{20}$$

It has been proved that computing a minimal feature subset from the whole attribute set is a NP-hard problem (Wong & Ziarko, 1985). Therefore, there are a lot of methods (Gao, Lai, Zhou, Zhao, & Miao, 2018; Jensen & Shen, 2004; Miao & Hu, 1999; Wang, Yu, & Yang, 2002) using the information gain (Eq. (1)) as the searching strategy to select the candidate feature. However, as is discussed above, the disadvantage of information gain is that it tends to select the attribute with more values, which has lower generalization capabilities for new samples in the classifier. Similar to C4.5, we choose the GainRatio (Eq. (4)) as the heuristic information to filter the candidate features. Algorithm 2 outlines the heuristic feature selection algorithm by using the quality of the rough approximation.

Algorithm 2 The heuristic feature selection algorithm based on the quality of rough approxiamtion.

Input: $\mathcal{D} = \langle X, \boldsymbol{y}, A, V\rangle$: training dataset;
Output: F: the selected feature subset, $\gamma_F(D)$: QRA.

```
C ← the attribute set of X;
F ← ∅;
repeat
    for a ∈ C − F do
        GR[a] ← GainRatio(D, a); //using Equation (4).
    end for
    a' ← arg max_a GR[a];
    F ← F ∪ a';
until (POS_F(D) == POS_C(D))
output F, γ_F(D);
```

5.1. Two-stage VO-kNN classifier

Since the feature selection algorithm and the VO-*k*NN classifier are constructed, we put it all together to establish the two-stage VO-*k*NN (TSVO-*k*NN) classifier. In the first stage, we employ Algorithm 2 to filter out the redundant attributes. Then, we use the output of the first stage as the input for Algorithm 1. Algorithm 3 outlines the details of the TSVO-*k*NN prediction algorithm as follows.

Algorithm 3 The two stage VO-*k*NN prediction algorithm.

Input: $\mathcal{D} = \langle X, \boldsymbol{y}, A, V\rangle$: training dataset, k: the number of nearest neighbors, $\boldsymbol{x}'$: a query sample;
Output: y: the class label of $\boldsymbol{x}$.

```
F ← the output of the Algorithm 2;
X' ← X with the attribute subset F;
A' ← F;
calculate the V' of X';
D' ← ⟨X', y, A', V'⟩;
let D' as the input of Algorithm 1;
y ← the output of Algorithm 1;
output y;
```

5.2. Complexity analysis

Compared with *k*NN classifiers, the computational complexity of the classification model VO-*k*NN is mainly focused on the calculation of VOHM, because the rest computation complexity of VO-*k*NN equals to *k*NN classifiers. Suppose N is the total number of objects in an information table, D is the total number of features, and C is the total number of class label, then the computational complexity of VOHM is $O(DCM)$, because the complexity of VLDM is $O(CM)$. As to the TSVO-*k*NN, the complexity should include the calculation of the feature selection part, which complexity is $O(DM)$. Now, suppose the cardinality of the selected attribute set is S, then the complexity of TSVO-*k*NN becomes to $O(DM + DCS) \approx O(DCS)$. Obviously, if the cardinality of the selected attribute set is small than D, then the TSVO-*k*NN classifier is more efficient than the VO-*k*NN classifier.

6. Experiments

The empirical study of the VO-*k*NN and TSVO-*k*NN classifier is given in this section. We first set up the experiments by introducing the datasets and comparison methods. Then we evaluate the

Table 3
Description of the UCI data sets.

Data set	Abbreviation	Attributes	Classes	Size
Weather	D1	5	2	14
Zoo	D2	17	7	101
Soybean	D3	36	4	47
Dermatology	D4	35	6	366
Lymphography	D5	19	2	148
Breast	D6	10	2	286
Balance	D7	5	3	625
Vote	D8	17	2	435

performance in terms of the prediction accuracy compared with other methods. Besides, we also evaluate the effects of factors such as parameter k and the selected attributes on classification performance.

6.1. Experimental setup

6.1.1. Data sets

To test the performance of our methods, we selected eight datasets with categorical values, i.e., Weather, Zoo, Soybean(small), Dermatology, Lymphography, Breast-cancer (abbr. Breast), Balance and Vote, which are collected from UCI machine learning responsory. For the sake of convenience, we use D1-D8 instead of the previous data sets. Table 3 lists the statistics for the training data sets.

6.1.2. Comparison methods and evaluation metric

We used the following methods for the experiments, most of them being introduced in Section 2, including our approach:

- *OL-kNN*. This kNN model simply classify objects with categorical attributes by using the overlap measure (Eq. (10)).
- *LE-kNN* (Chen & Guo, 2015). This model assumes that each attribute has a different weight. It uses attribute weighting methods based on local information entropy and applies these weights to overlap measure.
- *GE-kNN* (Chen & Guo, 2015). Unlike *LE-kNN*, the *GE-kNN* model employs the global information entropy to weighting attribute. Then the model embeds these weights into overlap measure to evaluate the distance between the categorical objects and to predict the class label of the object.
- *C4.5* (Quinlan, 1993). This model builds a decision tree to classify the categorical objects, according to the information gain rate.
- *Naive Bayes(NB)* (Hand & Yu, 2001; Murphy, 2012). This model assumes that the attributes are independent and need training data to fit the parameters of the model. The trained model will directly output the class label for the new object without comparing it with the neighbors.
- *VO-kNN*. This is the VOHM embeded k nearest neighbor classifier proposed for the categorical data in this paper.
- *TSVO-kNN*. This is two stages *VO-kNN* which performs feature selection task in the first stage and execute the classification task in the second stage.

In the experiments, we use the accuracy metric to evaluate the performance of the models above.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}. \tag{21}$$

where TP is the true positive, TN is the true negative, FP is the false positive and FN is the false negative.

6.2. Results

Table 4 demonstrates the results of the accuracy over the eight UCI data sets, respectively. All the results are performed under the settings with the 10-fold cross validation strategy and the parameter k range from 1 to 6. And, the average performances were reported in the format $\mu \pm \sigma^2$, where μ is average and σ^2 is the deviation.

The table shows that the two new classifiers (VO-kNN, TSVO-kNN) completely outperform the OL-kNN classifier, especially the accuracy of OL-kNN is zero on D1, D2, and D3 respectively. The reason is that the overlap measure does not take into account the potential relationship between different values in the attribute, and the information loss caused by the comparison strategy of overlap measure is too large so that the classification performance of OL-kNN is unreliable.

Both VO-kNN and TSVO-kNN have a better accuracy result than the attribute weighted model LE-kNN and GE-kNN, except the data set D2. Although the attribute weighted model has a good effect on the data set D2, the calculation of the attribute weights based on the local entropy(LE) and the global entropy(GE) that is employed by LE-kNN and GE-kNN respectively, has a significant impact on the performance of the classifier. For example, the performance of LE-kNN and GE-kNN on the data set D7 is 0.3837 ± 0.1528 and 0.6611 ± 0.0078, respectively. However, the performance of LE-kNN and GE-kNN on the data set D2 is 0.9736 ± 0.0173 and 0.9356 ± 0.0136, respectively. The local entropy and global entropy have their advantages on different data sets, but the core issue of the overlap measure that capturing the inherent order of the categorical attribute remains unresolved. It also demonstrates that VOHM considering the potential order relationship can better evaluate the distance between the categorical attribute values. Therefore, as can be seen from the table, VOHM embedded kNN classifiers, i.e., VO-kNN and TSVO-kNN, have better classification performance than the attribute weighted classifier on the data set D1 and D3-D8.

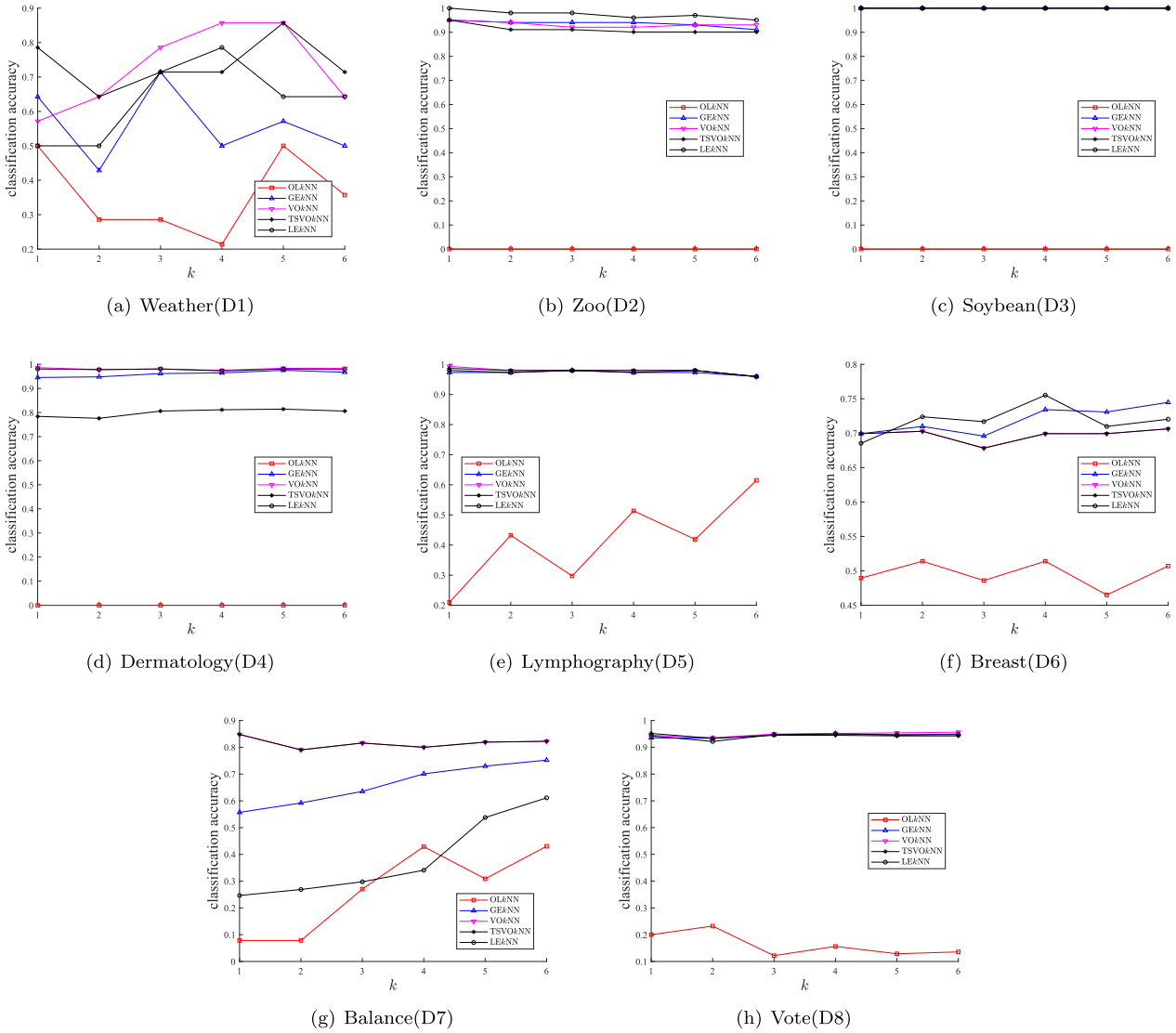
The performance of the C4.5 classifier is only best on the data set D6, and the rest of the results are not as good as VO-kNN and TSVO-kNN. Especially, the classification performance of the VO-kNN exceeds C4.5 by 32.12% and 18.4% on the data set D5 and D7. It is easy to find that the C4.5 classifier is suitable for data sets with few attributes and a few numbers of the categorical attribute values such as D6. When these conditions are not met, the performance of C4.5 will drop significantly. This is one of the reasons why C4.5 does not perform as well as VO-kNN and TSVO-kNN on the other data sets, especially on high dimensional data sets. It can be seen that the performance of C4.5 on the data set D1 is $0.5000 \pm 0.35.81$, which means that the algorithm is equivalent to random guess, which also shows that the VO-kNN and TSVO-kNN are more reliable than C4.5 from the Table 4.

Similarly, the classification performance of NB is only better than VO-kNN and TSVO-kNN on data set D7, while the performance on the other data sets is not as good as VO-kNN and TSVO-kNN. On the eight data set, the maximum value of the classification performance that the VO-kNN and TSVO-kNN exceeds NB is 16.67%, while the minimum amount is -2.83%. When the attributes of the data set are relatively independent, and the class distribution of the data set is relatively uniform such as D7, the performance of NB is better than the VO-kNN and TSVO-kNN. However, when these conditions are not satisfied, the VO-kNN and TSVO-kNN performs better than NB.

In summary, the VO-kNN and TSVO-kNN which can capture the inherent ordered relationship between categorical objects could obtain the intrinsic characteristics of the data, so that it is more conducive to improve the classification performance for the categorical data.

Table 4
The accuracy result of the models performed on the eight data sets.

Dataset	Accuracy						
	OL-*k*NN	LE-*k*NN	GE-*k*NN	VO-*k*NN	TSVO-*k*NN	C4.5	NB
D1	0.3571 ± 0.1195	0.6310 ± 0.1144	0.5595 ± 0.1051	0.7262 ± 0.1230	**0.7381 ± 0.0738**	0.5000 ± 0.3581	0.5714 ± 0.2417
D2	0.0000 ± 0.0000	**0.9736 ± 0.0173**	0.9356 ± 0.0136	0.9323 ± 0.0116	0.9125 ± 0.0192	0.8955 ± 0.0189	0.9089 ± 0.0105
D3	0.0000 ± 0.0000	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	**1.0000 ± 0.0000**	0.9710 ± 0.0106	**1.0000 ± 0.0000**
D4	0.0000 ± 0.0000	0.9786 ± 0.0032	0.9604 ± 0.0116	**0.9813 ± 0.0040**	0.7996 ± 0.0157	0.9316 ± 0.0176	0.9658 ± 0.0054
D5	0.4144 ± 0.1456	0.9741 ± 0.0079	0.9718 ± 0.0066	**0.9786 ± 0.0108**	0.9775 ± 0.0092	0.6574 ± 0.0236	0.9196 ± 0.0106
D6	0.4959 ± 0.0193	0.7185 ± 0.0226	0.7191 ± 0.0202	0.6976 ± 0.0098	0.6976 ± 0.0098	**0.7552 ± 0.1870**	0.7202 ± 0.0021
D7	0.2659 ± 0.1586	0.3837 ± 0.1528	0.6611 ± 0.0786	0.8160 ± 0.0199	0.8160 ± 0.0199	0.6320 ± 0.1842	**0.8398 ± 0.0811**
D8	0.1625 ± 0.0443	0.9425 ± 0.0104	0.9437 ± 0.0075	**0.9487 ± 0.0075**	0.9433 ± 0.0059	0.9440 ± 0.0306	0.9011 ± 0.0886



(a) Weather(D1) (b) Zoo(D2) (c) Soybean(D3)

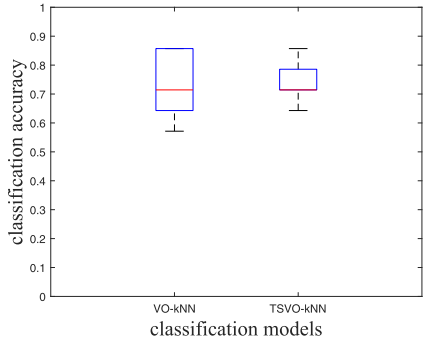
(d) Dermatology(D4) (e) Lymphography(D5) (f) Breast(D6)

(g) Balance(D7) (h) Vote(D8)

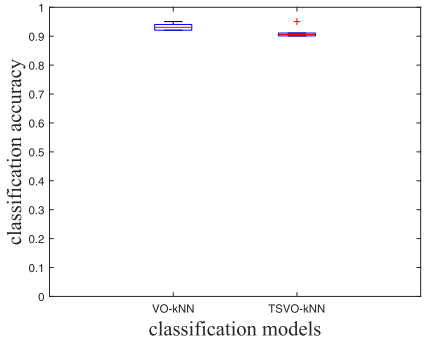
Fig. 2. The impact of the factor *k* for the family of the *k*NN classifiers.

6.3. *The effect of the parameter* k

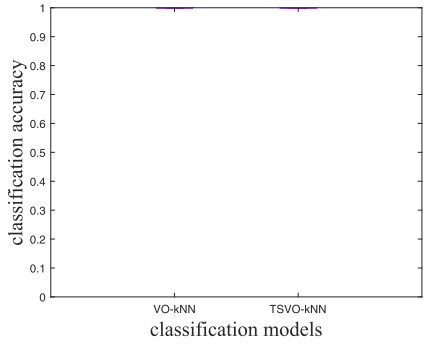
The parameter *k* is a key factor for the family of the *k*NN classifier. Fig. 2 shows the trend of the classification accuracy with *k* varying, which is generated by the OL-*k*NN, LE-*k*NN, GE-*k*NN, VO-*k*NN and TSVO-*k*NN classifier respectively. In Fig. 3 (b)–(e) and (h), the change of the classification accuracy is not obvious for all of the *k*NN based classifiers. In other words, the parameter *k* is not sensitive in the processing of the *k*NN based classification on the data set D2-D5 and D8, except the OL-*k*NN classifier on the data set D5 and D8. It is not worthy of discussing the effect of the parameter *k* for OL-*k*NN, because the classifier is not reliable for categorical data. In Fig. 3 (f) and (g), the fluctuation of the line produced by the VO-*k*NN and TSVO-*k*NN is relatively stable, while the other lines which are produced by the OL-*k*NN, LE-*k*NN and GE-*k*NN classifier are relatively large. In Fig. 3 (a), the value of *k* has a great influence on the classification accuracy of all classifiers. The main reason is that the distance calculated by the overlap



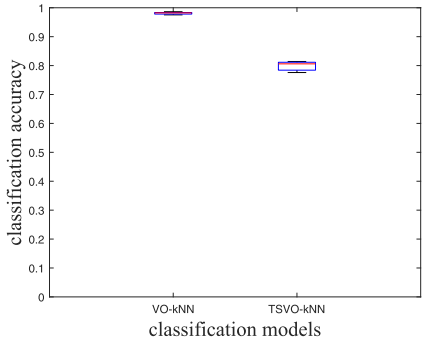
(a) Weather(D1)



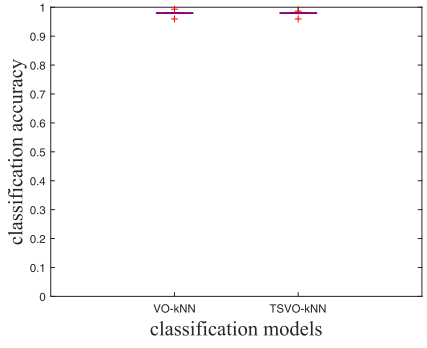
(b) Zoo(D2)



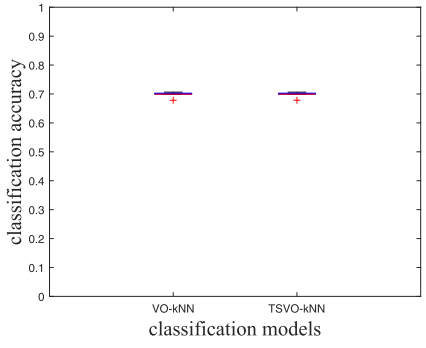
(c) Soybean(D3)



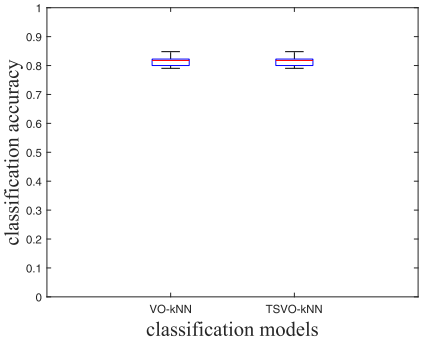
(d) Dermatology(D4)



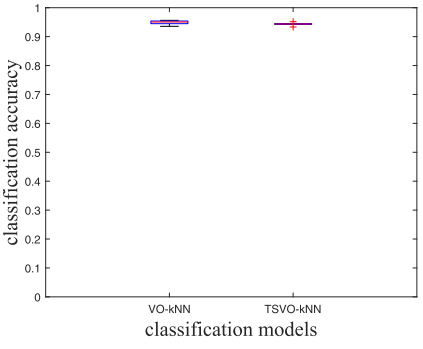
(e) Lymphography(D5)



(f) Breast(D6)



(g) Balance(D7)



(h) Vote(D8)

Fig. 3. The box plot of the accuracy w.r.t. VO-*k*NN and TSVO-*k*NN.

Table 5
Description of the selected attributes.

Data set	Abbr.	Condition attributes	Selected attributes	Proportion
Weather	D1	4	3	75.00%
Zoo	D2	16	5	31.25%
Soybean	D3	35	2	5.71%
Dermatology	D4	34	6	17.65%
Lymphography	D5	18	3	16.67%
Breast_cancer	D6	9	9	100.00%
Balance	D7	4	4	100.00%
Vote	D8	16	9	56.25%

measure, weighted overlap measure, and value-object hierarchy measure is so centralized that the uncertain of the neighbors of the object is too high. Furthermore, the uncertainty of the classification accuracy is brought by the concentrated character of the data. Therefore, the parameter k plays an important role in these types of data.

In summary, the parameter k indeed plays an essential role in the kNN based classifier for categorical data. The root reason is that the performance of the kNN based classifier is decided by the inherent characteristic of the data distribution. Although the situations the classifier would confront in the real applications, the VO-kNN and TSVO-kNN classifier are still less sensitive for the parameter k than the other classifiers in the categorical data environment, because the VOHM could capture the latent order relationship of object pairs from data sets.

6.4. The impact of feature selection

To verify the impact of attribute reduction on the eight data sets, we collected all the test results of the classification accuracy with the settings $k = 1.6$. The statistic of the selected attributes of the eight UCI datasets is described in Table 5.

From the table, we could see that The feature selection algorithm has a large number of compressed attributes on these datasets except on D6 and D7. And, the minimum cardinality of the selected attributes is 2, which means that the chosen attributes are only 5.71% of the original attributes.

Then, we analyze the effect of the selected attributes on these datasets. The box plot of the comparison result of the VO-kNN and TSVO-kNN is shown in Fig. 3. As can be seen from the figure, the selected attributes have the same ability of classification as the original attributes on the dataset D3, D5, D6, D7, and D8. However, the classification accuracy performed on the dataset D4 drops by 18.9% on average. But, the accuracy also is improved on the dataset D1. It means that the feature selection can compress the attributes and keep a better result than the original attributes. Although the feature selection would decrease the representation ability of the datasets, it also worth to add this stage to the VO-kNN classifier. In the worst case, if the result of the accuracy drops sharply, we could give up this stage because the data may have inherent complexity characteristic.

7. Conclusion

Due to the complexity of the data and the lack of effective non-numerical distance metrics, it is a challenging task of how to exploit the inherent characteristic of non-numerical data and to effectively represent it. In this work, we developed a novel value-object hierarchical metric to capture the latent order relationship from non-numerical data. And, we equip up nearest neighbor classifier with the new non-numerical metric and rough set theory based feature selection. It extends nearest neighbor classifier to be a more robust, representative, and effective model. Experiments show the validity of the model and that both VO-kNN and TSVO-kNN are more effective than the existing classifiers for non-numerical data.

As is mentioned before, non-numerical data is a commonly used data type in expert systems. The complexity of non-numerical data increases the difficulty of the algorithms performing the classification task. How to narrow the gap between classification algorithms and humans plays a crucial role to enhance the performance of algorithms. To solve the problem, several aspects of new models are worth investigating in depth. 1).*valid distance measures with more inherent relationships*. Roughly speaking, objects is a mixture representation of several non-numerical features. There may exist more complex interaction between attributes. Except for the relationship that is calculated by VOHM, we think of there should exist more complex relationships between non-numerical values such as the relationships between two group attribute values, the relationships between the values in the same group attributes etc. The more inherent characteristic of non-numerical data is founded, the more valid and effective the algorithm is. 2).*more effective classification models*. Although the kNN is an effective classification model to solve non-numerical classification, there also exist a lot of classification models such as SVMs, decision tree, Naive Bayes, etc. could be used in expert systems by incorporating the complex relationship that is computed by VOHM. How to use these classification models in non-numerical data environment indeed need to study in the future work. 3).*the strategy of feature selection*. How to select features which play crucial roles for non-numerical classification is the other problem should be taken into account in the future work. Reducing the redundant feature could minimize the scale of data and running time of algorithms. Therefore, the feature selection strategy should be further explored.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Sheng Luo: Writing - original draft, Writing - review & editing, Conceptualization, Formal analysis, Methodology, Data curation, Validation. **Duoqian Miao:** Conceptualization, Formal analysis, Supervision, Writing - review & editing, Funding acquisition. **Zhifei Zhang:** Writing - review & editing, Data curation, Investigation. **Zhihua Wei:** Validation, Writing - review & editing.

Acknowledgments

The authors thank both the editors and the anonymous referees for their valuable suggestions, which substantially improved this paper. This work is supported by National Key R&D Program of China (Grant no. 213), the National Science Foundation of China (Grant nos. 61673301, 61906137).

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th international conference on very large data bases*. In *VLDB '94* (pp. 487–499). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..

Alamelu, M. J., Milind, W. S., & Santhosh, K. V. (2013). A novel web page classification model using an improved k nearest neighbor algorithm. In 3rd international conference on intelligent computational systems (ICICS'2013) april (pp. 29–30).

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798–1828. doi:10.1109/TPAMI.2013.50.

Boriah, S., Chandola, V., & Kumar, V. (2008). Similarity measures for categorical data: A comparative evaluation. In Proceedings of the 8th SIAM international conference on data mining (pp. 243–254). SIAM.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. doi:10.1023/A:1010933404324.

Breiman, L. I., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees (CART). *Encyclopedia of Ecology, 40*(3), 582–588.

Buttrey, S. E. (1998). Nearest-neighbor classification with categorical variables. *Computational Statistics & Data Analysis, 28*(2), 157–169. doi:10.1016/S0167-9473(98)00032-2.

Cao, L. (2017a). Data science: A comprehensive overview. *ACM Computing Surveys, 50*(3), 43:1–43:42. doi:10.1145/3076253.

Cao, L. (2017b). Data science: Challenges and directions. *Communications of the ACM, 60*(8), 59–68. doi:10.1145/3015456.

Catanzaro, B., Sundaram, N., & Keutzer, K. (2008). Fast support vector machine training and classification on graphics processors. In *Proceedings of the 25th international conference on machine learning*. In *ICML '08* (pp. 104–111). New York, NY, USA: ACM. doi:10.1145/1390156.1390170.

Chen, L., & Guo, G. (2015). Nearest neighbor classification of categorical data by attributes weighting. *Expert Systems with Applications, 42*(6), 3142–3149.

Chen, L., Ye, Y., Guo, G., & Zhu, J. (2016). Kernel-based linear classification on categorical data. *Soft Computing, 20*(8), 1–13.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. doi:10.1007/BF00994018.

Cost, S., & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning, 10*(1), 57–78. doi:10.1007/bf00993481.

Daelemans, W., & Van den Bosch, A. (2005). *Memory-based language processing*. Cambridge University Press.

Gao, C., Lai, Z., Zhou, J., Zhao, C., & Miao, D. (2018). Maximum decision entropy-based attribute reduction in decision-theoretic rough set model. *Knowledge-Based Systems, 143*, 179–191. doi:10.1016/j.knosys.2017.12.014.

Garc A-Laencina, P. J., Sancho-G, M., Luis, J., Figueiras-Vidal, A., Bal, R., & Verleysen, M. (2009). K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputing, 72*(79), 1483–1493.

Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., & Yang, H. (2019). A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications, 115*, 356–372. doi:10.1016/j.eswa.2018.08.021.

Han, J., Pei, J., & Yin, Y. (2000). Mining frequent patterns without candidate generation. In Proceedings of the 2000 ACM SIGMODinternational conference on management of data (pp. 1–12). New York, NY, USA: ACM. doi:10.1145/342009.335372.

Hand, D. J., & Yu, K. (2001). Idiot's Bayes – Not so stupid after all? International Statistical Review, *69*(3), 385–398.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition: 1* (pp. 278–282). doi:10.1109/ICDAR.1995.598994.

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics, 15*(3), 651–674. doi:10.1198/106186006X133933.

Hu, Q., Yu, D., & Xie, Z. (2008). Neighborhood classifiers. *Expert Systems with Applications, 34*(2), 866–876.

Jahromi, M. Z., Parvinnia, E., & John, R. (2009). A method of learning weighted similarity function to improve the performance of nearest neighbor. *Information Sciences, 179*(17), 2964–2973.

Jensen, R., & Shen, Q. (2004). Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches. *IEEE Transactions on Knowledge and Data Engineering, 16*(12), 1457–1471. doi:10.1109/tkde.2004.96.

Jia, X., Shang, L., Zhou, B., & Yao, Y. (2016). Generalized attribute reduct in rough set theory. *Knowledge-Based Systems, 91*(C), 204–218. doi:10.1016/j.knosys.2015.05.017.

Jiang, L., Cai, Z., Wang, D., & Zhang, H. (2012a). Improving tree augmented naive Bayes for class probability estimation. Knowledge-Based Systems, *26*, 239–245.

Jiang, L., Cai, Z., Wang, D., & Zhang, H. (2014). Bayesian citation-KNN with distance weighting. International Journal of Machine Learning and Cybernetics, *5*(2), 193–199.

Jiang, L., Zhang, H., Cai, Z., & Wang, D. (2012b). Weighted average of one-dependence estimators. *Journal of Experimental & Theoretical Artificial Intelligence, 24*(2), 219–230.

Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 29*(2), 119–127.

Kim, H., & Loh, W.-Y. (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association, 96*(454), 589–604. doi:10.1198/016214501753168271.

Kim, K. (2016). A hybrid classification algorithm by subspace partitioning through semi-supervised decision tree. *Pattern Recognition, 60*, 157–163.

Kononenko, I. (1991). Semi-naive Bayesian classifier. In *Proceedings of the european working session on learning on machine learning*. In *EWSL-91* (pp. 206–219). Berlin, Heidelberg: Springer-Verlag.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. doi:10.1038/nature14539.

Li, H., Wen, G., Yu, Z., & Zhou, T. (2013). Random subspace evidence classifier. *Neurocomputing, 110*, 62–69. doi:10.1016/j.neucom.2012.11.019.

Liu, C., Cao, L., & Yu, P. S. (2014). Coupled fuzzy k-nearest neighbors classification of imbalanced non-IID categorical data. In *International joint conference on neural networks* (pp. 1122–1129).

Loh, W.-Y. (2009). Improving the precision of classification trees. *The Annals of Applied Statistics, 3*(4), 1710–1737. doi:10.1214/09-AOAS260.

Loh, W. Y. (2012). *Classification and regression trees*. John Wiley & Sons, Ltd.

Miao, D., & Hu, G. (1999). A heuristic algorithm for reduction of knowledge. *Journal of Computer Research & Development, 36*(6), 681–684. (In Chinese).

Morlini, I., & Zani, S. (2012). A new class of weighted similarity indices using polytomous variables. *Journal of Classification, 29*(2), 199–226.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.

Mller, P., Salminen, K., Nieminen, V., Kontunen, A., Karjalainen, M., Isokoski, P., … Surakka, V. (2019). Scent classification by k nearest neighbors using ion-mobility spectrometry measurements. *Expert Systems with Applications, 115*, 593–606. doi:10.1016/j.eswa.2018.08.042.

Pawlak, Z. (1998). Rough set theory and its applications. *Journal of Telecommunications & Information Technology, 3*(3), 7–10.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..

Rodger, J. A. (2014). A fuzzy nearest neighbor neural network statistical model for predicting demand for natural gas and energy cost savings in public buildings. *Expert Systems with Applications, 41*(4, Part 2), 1813–1829. doi:10.1016/j.eswa.2013.08.080.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: A modern approach*. Malaysia: Pearson Education Limited.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning.. *Communications of the ACM, 29*(12), 1213–1228. doi:10.1145/7902.7906.

Walker, S. H., & Duncan, D. B. (1967). Estimation of the probability of an event as a function of several independent variables. Biometrika, *54*(1–2), 167–179.

Wang, G. Y., Yu, H., & Yang, D. C. (2002). Decision table reduction based on conditional information entropy. *Chinese Journal of Computers, 25*(7), 759–766. (In Chinese)

Wei-Yin, L. (2014). Fifty years of classification and regression trees. *International Statistical Review, 82*(3), 329–348. doi:10.1111/insr.12016.

Wong, S. K. M., & Ziarko, W. (1985). On optimal decision rules in decision tables. Bulletin of the Polish Academy of Sciences Mathematics, *33*(11–12), 693–696.

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering, 12*(3), 372–390. doi:10.1109/69.846291.

Zhao, H., & Li, X. (2017). A cost sensitive decision tree algorithm based on weighted class distribution with batch deleting attribute mechanism. *Information Sciences, 378*, 303–316.

Zheng, F., & Webb, G. I. (2011). *Tree augmented naive Bayes*. US: Springer.