An Improved kNN Algorithm Based on Conditional Probability Distance Metric

Liu Ziyang[1,a], Gao Zhanbao[1,b,*], Li Xulong[1,c]

[1] School of Automation and Electrical Engineering, Beihang University, XueYuan Road No.37,HaiDian District, BeiJing,China

[a] 2210903484@qq.com, [b] gaozhanbao@buaa.edu.cn, [c] xulongli@yeah.net

Keywords: kNN, Nominal Variable, Distance Metric, Conditional Probability, QPSO

Abstract. There is no rational distance metric for nominal variables in traditional kNN classification algorithm. And the weighting methods commonly used in kNN cannot process datasets with multi-type variables or depend much on the field knowledge. An improved kNN method based on conditional probability and QPSO is presented is this paper. This approach measures the distance between two nominal variables by the distribution difference of instances' classes, which makes full use of attribute values' information. Meanwhile, it adopts QPSO to adjust attribute weight so that the weight will enhance the classification accuracy. This approach is able to process datasets with multi-type variables and less depends on parameters. Finally, experiments were taken on the UCI data set, which shows that our approach is superior in performance to algorithms compared.

1. Introduction

kNN algorithm(k-Nearest Neighbour, kNN) is a well-known pattern recognition and statistical method proposed by Cover and Hart. It is widely used in text and image classification, fault diagnosis and disease diagnosis. Similarity metric and attribute weight is the key of kNN. They determine the quality of test instances' neighbours and affect the classifier's classification accuracy.

Euclidean distance is the commomly used distance metric in traditional kNN. But for nominal variables, people usually set the local distance as 0 when the two instances have the same value, otherwise 1. In [1], gray correlation is used to calculate the similarity of two instances, which can only be applied to the case where there is a significant correlation between the instances' attribute trend and their classes. In [2], the mean value of attribute information entropy is used as the distance between the two instances. In [3], a method that employes the semantic distance to represent the instances distance was proposed.

Weighting approach is a popular topic in many fields[4]. And what is commonly used in kNN is Delphi which is intuitive and well-understood but depends on the domain experts. Statistical analysis based methods, such ascorrelation, Pearson correlation coefficient. They get attribute weight by calculating correlation between the instances'attribute value and their class. Entropy weighting method[5] based on the variation of the attribute value. In addition, there are another weighting methods adopting machine learning algorithm, such as rough set[6].

In practical application, such as disease diagnosis, classification on data contains nominal variables, numeric variables and ordinal variables is the often case. To improve the performance of kNN in multi-type data, we propose PkNN+QPSOW which measures the instances similaity measure based on conditional probability and obtains attribute weight by QPSO (Quantum-Behaved Particle Swarm Optimization).

2. Problem Definition and Traditional kNN

The training set $X = \{X_1\ X_2\ X_3 ... X_N\}$ consists of N instances and the instances are divided into n groups. X contains m attributes denoted as $A_1, A_2, ..., A_M$, so each instance is an m-dimensional vector. The value of X_i ($i \in [1, N]$) on A_j ($j \in [1, M]$) is $A_{ij}^{'}$ and $\max(A_j^{'})$, $\min(A_j^{'})$ is the maximum and minimum

value of A_j. Correspondingly A_{ij} is the normalized value of $A_{ij}^{'}$ by formula (1). In traditional kNN, the distance between target instance X_t and training instance X_i is calculated by formula (2) and formula (3).

$$A_{ij} = \frac{A_{ij}^{'} - \min(A_j^{'})}{\max(A_j^{'}) - \min(A_j^{'})} \tag{1}$$

$$D(X_i, X_t) = \sum_{j=1}^{M} w_j D_j^{\text{local}}(X_i, X_t), i \in [1, N], j \in [1, M] \tag{2}$$

$$D_j^{\text{local}}(X_i, X_t) = \begin{cases} (A_{ij} - A_{it})^2, & A_j \text{ is continuous or ordinal} \\ 1, & A_j \text{ is nominal and } A_{ij} \neq A_{it} \\ 0, & A_j \text{ is and } A_{ij} = A_{it} \end{cases} \tag{3}$$

Where w_j is the weight of A_j, and it subjects to $\sum_{j=1}^{M} w_j = 1$. The distance between X_i and X_i is the weighted sum of D_j^{local}.

The weight calculation based on information entropy is as follows:

$$p_{ij} = A_{ij} \Big/ \sum_{i=1}^{N} A_{ij} \quad i \in [1, N], j \in [1, M] \tag{4}$$

$$e_j = -k \sum_{i=1}^{N} p_{ij} \cdot \ln p_{ij} \quad , k = 1/\ln N \tag{5}$$

The weight of A_j:

$$w_j = (1 - e_j) \Big/ \sum_{j=1}^{M} (1 - e_j) \tag{6}$$

The training instances are ranked by the computed distance in ascending order and the first *k* instances will be taken as neighbours. Then, the most predominant class label will be assigned to the target instance. k may influence the performance and the noise tolerance of kNN[7].

3. PkNN+QPSOW

In the 0 or 1 approach, the nominal variables' distances tend to cover the difference of other variables'. We propose a similarity measure method based on attribute conditional probability. For nominal variable, since it cannot be measured numerically, may wish to change the perspective on the problem, considering the relationship between attribute values and instances categories. As shown in Fig.1, given different attribute values, the distribution of the instance class is different in shape. Fig.1 shows the conditional probability distribution (histogram) for the three different attribute values which are nominal (the data set is divided into 5 groups). To show the distribution difference visually, the conditional probability values are curve fitted (curve in Fig.1).

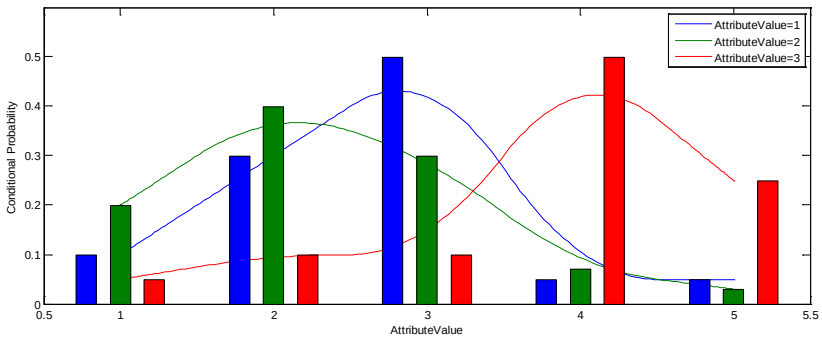

Fig.1 The conditional probability distribution of samples under different attribute values

The distribution difference reflects the attribute value difference from the side, so we employ the distribution difference to determine the similarity between different attribute values. The distance measure model based on conditional probability:

$$D(X_i, X_t) = \sum_{j=1}^{M} w_j D_j^{\text{local\_p}}(X_i, X_t), i \in [1, N], j \in [1, M] \tag{7}$$

$$D_j^{\text{local\_p}}(X_i, X_t) = \begin{cases} |A_{ij} - A_{tj}|, & A_j \text{ is continuos or ordinal} \\ \frac{1}{C}\sum_{k=1}^{C} |P(C_k | A_{ij}) - P(C_k | A_{tj})|, & A_j \text{ is nominal} \end{cases} \tag{8}$$

where $D_j^{\text{local\_p}}$ is the local distance between X_i and X_t on A_j.

It can be seen that the larger the difference in sample distribution is the larger the local distance will be. In the extreme case where instances with different attribute values belong to two different categories, the local distance is 1. In addition, if the instances distribution is exactly the same, the distance is 0 according to Eq.(9). So, the similarity measure based on conditional probability describes the instance difference more reasonably and accurately.

We propose the attribute weighting method, called QPSOW (QPSO for Weighting), which obtains attribute weight by QPSO. QPSO is proposed to overcome the shortcomings of Particle Swarm Optimization (PSO), such as too many parameters, low randomness and no guarantee for global optimal. In an *M*-dimension search space, $w = \{w_1, w_2, ..., w_S\}$ represents the problem's potential solution. At time *t*, the position of w_i is $w_i(t) = [w_{i1}(t), w_{i2}(t), ..., w_{iM}(t)]$, $i \in [1, S]$. The iterative formula is as follows:

$$p_{ij}(t) = \varphi_j(t) w_{ij}^*(t) + [1 - \varphi_j(t)] G_j(t), \quad \varphi_j(t) \sim \text{U}(0,1) \tag{9}$$

$$w_{ij}(t+1) = p_{ij}(t) \pm \alpha |E_j(t) - w_{ij}(t)| \ln[1/u_{ij}(t)], \quad u_{ij}(t) \sim \text{U}(0,1) \tag{10}$$

Where subscript *i* represents the *i*-th particles, *j* represents its *j*-th dimension, $w_i^*(t) = [w_{i1}^*(t), w_{i2}^*(t), ..., w_{iM}^*(t)]$ is the best place for $w_i(t)$, $G(t) = [G_1(t), G_2(t), ..., G_M(t)]$ is the group's global best place, and $E(t)$ is the average optimal position for the particle swarm.

$$\alpha = \alpha_{\max} - \frac{t}{Iteration}(\alpha_{\max} - \alpha_{\min}) \tag{11}$$

Where $\alpha_{\max}$ and $\alpha_{\min}$ are the maximum and minimum values respectively.

QPSOW is an objective weighting method, which only depends on the training set. The weight is obtained by minimizing the objective function, as in equation (13) and (14); so that the weight is adjusted towards the direction where validation-set gets the maximum accuracy. To reduce the random error, we obtain *t*-group training set and validation set by random sampling. And we take the mean of all weights as the final weight, which can be seen in equation (15).

$$f(w) = \min \sum_{i=1}^{N_v} D_i(w) \tag{12}$$

$$D_i(w) = \begin{cases} -D(X_i, X_i^{near}), & X_i \text{ and } X_i^{near} \text{ belong to the same class} \\ D(X_i, X_i^{near}), & \text{otherwise} \end{cases} \tag{13}$$

$$w_j = \frac{1}{T}\sum_{k=1}^{T} w_j^k \tag{14}$$

X_i^{near} is the nearest neighbour to X_i, T is the experiments number, and w_j^k is the weight of A_j calculated at time *k*.

The detailed procedure of PkNN+QPSOW is as follows.

Input: Training set *X*; Target instance X_t; Number of nearest neighbour k; Number for learning times for learning attribute weight *T*

Output: Class of X_t.

(1). $X \leftarrow$ Remove instances with vacancies, and normalize in *X* and X_t by Eq.(1).

(2). $N(A_{ij}) \leftarrow$ Number of instance containing the attribute value A_{ij}.

(3). $N(C_k | A_{ij}) \leftarrow$ Number of instance which contains A_{ij} and belongs to the class C_k.

(4). $P(C_k | A_{ij}) \leftarrow$ Take the frequency $\hat{P}(C_k | A_{ij})$ calculated by in Eq.(15) as the $P(C_k | A_{ij})$.

(5). $D_i \leftarrow$ Obtain the nominal variable distance metric matrix by Eq.(9). D_i is shown in Eq.(16).

$$\hat{P}(C_k \mid A_{ij}) = \frac{N(C_k \mid A_{ij})}{N(A_{ij})} \tag{15}$$

$$D_i = \begin{pmatrix} D_{11}, D_{12}, \ldots, D_{1m} \\ D_{21}, D_{22}, \ldots, D_{2m} \\ \ldots \quad \ldots \quad \ldots \quad \ldots \\ D_{m1}, D_{12}, \ldots, D_{mm} \end{pmatrix} \tag{16}$$

Where m is the number of different attribute values of A_i, and D_{ij} is the local distance between the i-th and the j-th attribute values. Obviously D_i is symmetric matrix.

(6). For k=1:1:T

$S_v, S_t \leftarrow$Take one tenth of the training set as S_v by simple random sampling without replacement, and the rest as S_t.

$w^k \leftarrow$Get $f(w^k)$ by Eq.(12) and Eq.(13), and execute QPSO algorithm to minimize it.

End for

$w \leftarrow$ Get the final attribute weight by Eq.(14).

(7). $C \leftarrow$Computes the distance between each training instance and target instance by Eq.(8) and Eq.(9). And assign it to the class mostly occurring among the k neighbours.

4. Experiments and Results

The data used in this research is UCI Mammographic Mass Data (MM) which is a medical data set containing 961 instances of which 516 are benign, 445 are malignant, and there are some instances with missing values[8].

Table 1 Formula and implication of performance measures.

Performance Measures	Formula	Meaning
Accuracy	(TP+TN)/(TP+TN+FP+FN)	Predictive accuracy is the performance measure generally associated with machine learning algorithms
Sensitivity	$\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$	True positive rate or accuracy of positive class
Specificity	$\mathrm{TN}/\mathrm{FP}+\mathrm{TN}$	True negative rate or accuracy of negative class

The performance of machine learning algorithms is typically evaluated by a confusion matrix (for a 2 class problem). Table 1 shows the relevant definitions of the performance measures.

The experiments are divided into two parts. In the first part, PkNN+QPSOW is compared with kNN+Entropy (Entropy weight kNN) and kNN+Delphi (Delphi weight kNN) with the varying k. We take accuracy, sensitivity and specificity as the performances measures of these methods. The weight of Delphi method comes from [9]. In the experiment, k is set as 3, 5, 7 respectively and T set as 10. 10-fold cross validation experiment was carried out10 times under different k. The experimental results are shown in Fig.2.

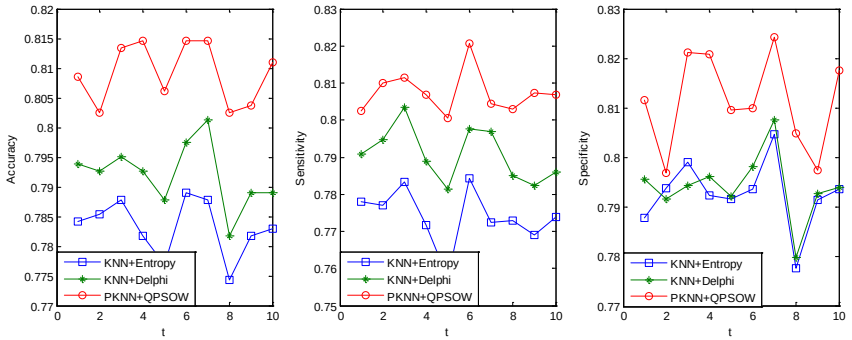

Fig.2 Comparison of the proposed method, kNN+Entropy and kNN+Delphi when k=3.

As can be seen from Fig.2, the classification accuracy of proposed method is higher than that of the kNN methods. At the same time, its sensitivity is better while the specificity is not less than that of the compared methods. With the increase of k, the performance measures of proposed method

changes smaller (while keeping ahead) than that of the compared methods, which indicates it less dependence on the parameters.

Local distance of nominal variable tends to dominate global distance between two instances when the traditional 0 or 1 approach is applied. So instances with the same nominal variable value are more likely to be neighbours, and the contribution of the remaining variables to the distance will be offset, resulting in improper selection of neighbours and false classification. The proposed method takes the instances' distribution and category information into account, which is actually the utilization of the deep information contained in the data, so it makes the nominal variable distance metric more reasonable.

Table 2 shows the mean value of 10 experimental results under different k. The mean value of our method is higher than that of the compared approaches, which indicates that the overall performance of our method is better.

Table 2 Mean value of the performance metrics for each method in 10 experiments.

Method	k=3			k=5			k=7		
	Acc	Sen	Spe	Acc	Sen	Spe	Acc	Sen	Spe
P*k*NN+QPSOW	**0.809**	**0.807**	**0.811**	**0.819**	**0.814**	**0.824**	**0.825**	**0.821**	**0.830**
*k*NN+Entropy	0.783	0.774	0.793	0.801	0.783	0.818	0.807	0.798	0.813
*k*NN+Delphi	0.792	0.791	0.794	0.803	0.786	0.819	0.812	0.798	0.827

Table 3 Mean value of the performance measures for each method in 10 experiments.

Method	Accuracy	Sensitivity	Specificity
P*k*NN+QPSOW	**0.825**	**0.821**	**0.830**
Decision tree	0.799	0.796	0.803
BP	0.794	0.797	0.791

The second part of the experiment compares the proposed method with decision tree (CART) and three-layer BP neural network (set the hidden layer as 100 empirically). In the same way, 10-fold validation is performed 10 times. The results are shown in Fig.3.

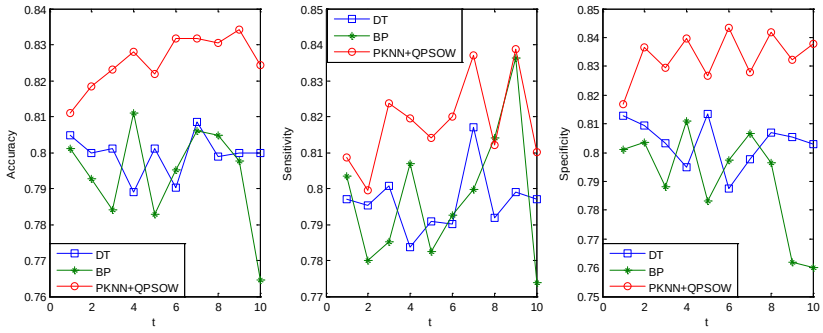

Fig.3 Comparison of the proposed method, DT and BP when k=7.

It can be seen from Fig.3 and table 3 that the performance of decision tree and the BP is almost. Compared with DT and BP, PkNN+QPSOW performance measures are about 2 percentage points higher. The results indicate that our method is not only superior to the same type kNN algorithm, but also comparable to other commonly used classification algorithms.

5. Conclusions

We propose PkNN method which uses the difference in instances distribution under different nominal variables to measure the local distance on the attribute. This approach makes full use of the distribution information of the instances under different attribute values. QPSOW makes use of QPSO's excellent global optimization ability and fast convergence speed, so that the attributes' weights are conducive to higher classification accuracy. Our method combines PkNN and QPSOW with kNN, and the experiment results show that it outperforms all the other four algorithms and has better stability. It should be noted that PkNN+QPSOW's false negative rate is less than the compared method, which is more important in the practical medical diagnosis setting.

References

[1] WANG Hong-yu, NI Zhi-wei, YAN Jun. Research on Application of Grey-Relational Theory in CBR,2010, 20(5):96-99. (in Chinese with English abstract).

[2] YANG Li, ZUO Chun, WANG Yu-Guo. K-Nearest Neighbor algorithm based on information entropy of attribute value. Computer Engineer and Application, 2010, 46(3):115-117. (in Chinese with English abstract).

[3] TONG Xian-Qun, ZHOU Zhong-mei. K-Nearest Neighbor Classification Based on Semantic Distance. Journal of Software, 2010, 46(3):115-117. (in Chinese with English abstract).

[4] Greene D, Freyne J, Smyth B, et al. An Analysis of Research Themes in the CBR Conference Literature[C]// Advances in Case-Based Reasoning, European Conference, Eccbr 2008, Trier, Germany, September 1-4, 2008. Proceedings. 2008:18-43.

[5] WANG Zeng-min, WANG Kai-jue. Improved KNN algorithm based on entropy method. Computer Engineer and Application, 2009, 45(30):129-131. (in Chinese with English abstract).

[6] LIU Ji-yu, WANG Qiang, LUO Chao-hui, et al. Weighted KNN Data Classification Algorithm Based on Rough Set. Computer Science, 2015, 42(10):281-286. (in Chinese with English abstract).

[7] Maillo J, Ramírez S, Triguero I, et al. kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors Classifier for Big Data[J]. Knowledge-Based Systems, 2016.

[8] M. Elter, R. Schulz-Wendtland, T. Wittenberg. mammographic-masses[DB/OL]. http://archive.ics.uci.edu/ml/machine-learning-databases/mammographic-masses/mammographic_masses. data, 2016-12-10/2016-12-20.

[9] Gu Dongxiao. Research on Clinical Diagnosis & Treatment Supporting Technology Based on Case Base[D]. HeFei University of Technology, 2011.