Wu Fenlin[1],
Zheng Yifei[1],
Wang Cheng[2]

1 – Xiamen Medical College, Xiamen, China
2 – HuaQiao University, Xiamen, China

ADAPTIVE NORMALIZED WEIGHTED KNN TEXT CLASSIFICATION BASED ON PSO

У Феньлінь[1],
Чжен Іфей[1],
Ван Чен[2]

1 – Сяминьський медичний колледж, м. Сяминь, КНР
2 – Університет ХуаЦяо, м. Сяминь, КНР

КЛАСИФІКАЦІЯ ТЕКСТУ АДАПТИВНИМ НОРМАЛІЗОВАНИМ ВЗВАЖЕНИМ МЕТОДОМ KNN НА ОСНОВІ ОПТИМІЗАЦІЇ МЕТОДОМ РОЮ ЧАСТОК

Purpose. Classical KNN text classifier has some shortcomings such as consistent weights of each characteristics, which causes low classification accuracy and high feature dimension that leads the program to run too much time when facing a large datasets. To solve these problems, a normalized feature weighted KNN text classifier was proposed (it was named NPSOKNN algorithm).

Methodology. The overall accuracy of classifier was used as the global optimization goal of feature weights. PSO was used to search the global optimization feature weights. In order to reduce the number of features and time cost of KNN text classifier, we set a threshold to drop the features that are lower than the threshold value.

Findings. We first got the global optimization feature weights, and then by using these weights and feature reduction method, we obtained a new feature vector, the dimension number of which is much smaller than the original text vector and with high accuracy of text classification.

Originality. We made a study of the improvement of the text classifier by using improved PSO and KNN. We discussed normalized feature weights, weighted distance calculation function, and feature dimension reduction. The research on this aspect has not been found at present.

Practical value. The 10-fold cross-validation experimental results showed that the average accuracy of NPSOKNN is higher than that of the classical KNN in text classifier, and the time cost was reduced significantly because of features reducing.

Keywords: *text classification, KNN, normalized feature weight, feature weight optimization, PSO, feature reduction*

Introduction. With the exponential growth of information, it is a challenge for information science about how to effectively organize and manage information, and to rapidly, accurately and comprehensively provide users with the desired information. Under this kind of background, automatic text classification technology becomes an important research area. KNN (K-Nearest Neighbor) is an instance-based classification method proposed by T.M. Cover and P.E. Hart [1]. The basic idea of KNN text classification is: firstly, to find k most closest texts out of a comprehensive set; then, identify the most frequent type from the k-most closest texts; categorize this text with this type. Among classification algorithms, KNN is a widely used text classifier because of its simplicity and efficiency. Some of KNN technology can theoretically achieve the same result that could be achieved by Bayesian decision making with complete prior. KNN can adapt to more complex distribution. For unknown and un-normal distribution, it can achieve higher classification accuracy.

However, KNN text classification also has some problems in classification, such as:

1. KNN is a lazy instance-based learning method. It defers the decision on how to generalize beyond the training data until each new query instance is encountered. So for high-dimensional samples or large sample set, the time of similarity computing is huge, making it unpractical [2].

2. Classic KNN classifier considers all features with the same weight. In other words, all the features take the same effect on similarity computing. However, in the real application, some features are strongly associated with the classification; some are weakly associated with the classification [3]. If the classifier does not distinguish between the contributions of each feature, its classification effect will be worse.

One method to solve these problems is based on rough set combined with other classifiers, such as multi-classifier fusion based on rough sets and support vector machine, Bayesian classifier based on rough set algorithm [4], maximum entropy text classification algorithm based on rough sets, and KNN text classification algorithm based on rough sets and so on. The classification algorithm based on fusion technology uses rough set to reduce feature dimension, and then uses classification algorithms for further processing. It is able to overcome the shortcomings of KNN text classifier when dealing with a high dimensional feature. Nevertheless, the computational complexity of rough set used in feature reduction is too high, prone to lead to "combinatorial explosion" problems [5].

Another popular algorithm is to use clustering technology to improve the efficiency of text classifier. Such algorithm easily leads to poor results due to the incorrectly set threshold, uneven distribution of training samples and other reasons.

Another type of improved algorithm is based on feature weighting. Such algorithms give different weights to each feature to distinguish the role of these features on classifier [6, 7], used semantic features and information entropy weighting to do feature clustering. Experimental results showed that, compared to TF-IDF, its classification accuracy rate increased by about 5% [8], calculated feature weight according to the number of feature word occur in a particular class, proposed a weighted KNN algorithm based on PSO, of which the main idea is to give greater importance to features with higher weight. It used PSO algorithm to search the optimal weights: test in the UCI datasets showed that weighted features not only improved classification accuracy but also reduced the feature dimensions. The main problem is unnormalized weight searching easily falls into local optimal value, because too large space results in hard-to-find optimal weights.

Another big challenge for automatic text classification algorithm is dimension: typically up to tens of thousands of dimensions, resulting in too much classification time. Therefore, to figure out more important dimensions for improving the accuracy and efficiency of classification has great significance. Typical feature selection methods are information gain (IG), mutual information (MI), document frequency (DF). However, in dealing with unbalanced data sets, these commonly used feature selection methods tend to choose the feature subset that benefits for large number class category, thus the classify effect of some small class category could be poor.

The algorithm normalizes the features weight, makes the maximum unique. Weight normalization diminishes the solution space, reduces multimodal problems, and speeds up peak problem processing. Uses average normalized weight as PSO's initial value. In order to avoid local optimum in optimization, the algorithm introduced mutation operator. Finally, the improved PSOKNN is applied to high-dimensional text classification. By comparing the experimental results, the classifier using cross method is more precise. In addition, in terms of time consuming we compared feature reduction and optimization time.

PSO adaptive normalized weighted KNN text classification (NPSOKNN algorithm). A solution is designed to solve the problems presented above, by making the precision of classification as optimization objective function and giving different weight to each feature, to improve the accuracy of the classifier.

NPSOKNN algorithm. Using the included angle cosine method as the text similarity measure in the classical KNN classifier, the similarity between unclassified sample feature vector x and classified sample vector d_i can be described as follows

$$sim(x,d_i)=\frac{\sum_{k=1}^{m}(\omega_k\times\omega_{ik})}{\sqrt{(\sum_{k=1}^{m}\omega_k^2)(\sum_{k=1}^{m}\omega_{ik}^2)}}\ . \qquad (1)$$

Where, m is the dimension of feature vector, ω_k is the k^{th} dimension value of vector x, ω_{ik} is the k^{th} dimension value of vector d_i, $k=1,2,...,\ m,\ i=1,\ 2,\ n,\ n$ is the total number of samples classified.

The formula to calculate category weight of the text to be classified

$$y(x,C_j)=\sum_{d\in knn} r_{ij}y(d_i,C_j)\ .$$

Where, x is the feature vector of the text to be classified, $y(d_i,C_j)\in\{0,1\}$ denotes whether document d_i belongs to the class C_j, r_{ij} denotes the similarity between x and d_i.

During calculating the distance among texts with all features, this paper introduces weight factor to discriminate relative strength of features and classification and removes weak relevant features according to weight, which can reduce the feature dimension and classification time while improving classification accuracy.

The weighted optimization objective function. This paper uses F_1 test value to synthesize the precision and recall, which are given the same importance to consider. The macro average of F_1 test value is as the weighted objective function

$$Macro-F_1^*=maximum\left(Macro-F_1\right).$$

In the above formula,

$$Macro-F_1=\frac{2\overline{r}\overline{p}}{\overline{r}+\overline{p}}\ ;$$

$$\overline{r}=\frac{\sum_{c_i\in C}a}{\sum_{c_i\in C}a+\sum_{c_i\in C}c}\ ;$$

$$\overline{p}=\frac{\sum_{c_i\in C}a}{\sum_{c_i\in C}a+\sum_{c_i\in C}b}\ .$$

Where p is the accuracy rate, r is recall ratio, a is selected relevant documents, b is selected irrelevant documents, and c is unselected relevant documents.

The advantages of the normalized weight.

1. Normalized feature weight used in NPSOKNN text classification can reduce the solution space and make the optimal solution unique. Fig. 1 hypothesis feature dimension as two dimensions and the solution space is reduced by half after normalization, which can be seen from the chart.

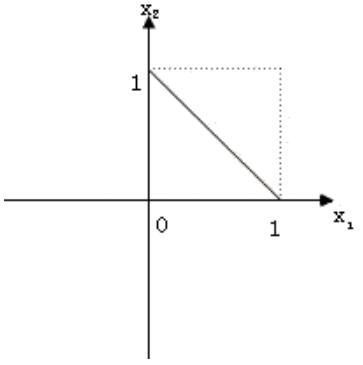


Fig.1. Schematic diagram of normalized solution space

2. Speed up to process multi peak problems. Fig. 2 shows that there will have an increase in artificial multi peak problem without normalizing weight. On the other hand, multi-

peak value problem can be translated into a single value problem after normalizing weight (Fig. 3).

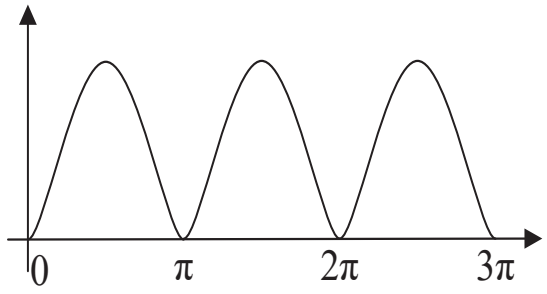


Fig. 2. Multi peak problem

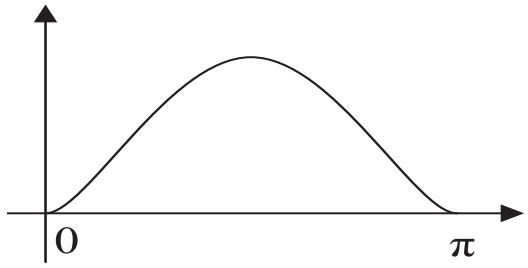


Fig. 3. Single value problem

3. Dimensionality reduction. Reduce the final dimension of the feature vector, which can be used as reference to judge whether a particle is legal:

The i^{th} particle $\omega_i = (\omega_{i1}, \omega_{i2}, ..., \omega_{im}) \in \mathbb{R}^m$,

$$\sum_{j=1}^{m} \omega_{ij} = 1, \omega_{i1}, \omega_{i2}, ..., \omega_{im} \in [0,1].$$

Thus

$$\omega_{im} = 1 - \sum_{j=1}^{m-1} \omega_{ij} \ . \qquad (2)$$

If ω_{ij} <0, then the i^{th} particle is illegal.

Using PSO algorithm to get the global optimization normalized feature weights adaptively. Thus, the key problem of the algorithm proposed in this paper is to get the optimal solution of feature weight. Simple and effective in seeking optimization, PSO is adopted to learn feature weight. In order to prevent particles from falling into local optimum, the mutation operator is introduced into PSO.

In 1995, Kennedy and Eberhar proposed PSO algorithm, inspired by the foraging behavior of bird flocks. The PSO optimization algorithm regards a solution to the optimization problem as the position of a bird in the searching space, which is called the particle. Each particle has a fitness value (candidate solution) determined by the optimal function and a velocity to decide its flying direction and range. In the optimization process, each particle remembers and follows the current optimal particle, then seeks the optimal solution in the solution space.

Particle swarm algorithm is described as follows, assuming the population size is *N*, during the *t*th iteration, the coordinate position of each particle in *D* dimensional space can be represented as $\overline{x_i}(t) = (x_i^1, x_i^2, ..., x_i^d, ..., x_i^D)$ and the velocity is represented as $\overline{v_i}(t) = (v_i^1, v_i^2, ..., v_i^d, ..., v_i^D)$. Then adjust the position and velocity for the next iteration in accordance with the following methods

$$\overline{v_i}(t+1) = \omega \overline{v_i}(t) + c_1 r_1 (\overline{p_i}(t) - \overline{x_i}(t)) + c_2 r_2 (\overline{p_g}(t) - \overline{x_i}(t)); \quad (3)$$

$$\overline{x_i}(t+1) = \overline{x_i}(t) + \overline{v_i}(t+1) \ .$$

The algorithm uses the optimization ability of particle swarm to find out the optimal weight that can centralized the characteristics for each feature. Then remove the small relative weight and join the weight into calculating the similarity of texts.

For *n* classes problems, assume the *i*th class $c_i (i=1,2,...,n)$ has N_i samples, and $x_j^{(i)} (j=1,2,...,N_i)$. A brief solution steps is given below:

Step 1: Extract *m* features from the $\sum_{i=1}^{n} N_i$ sample space, store them in an ordered set *F*, vectorize the training set and generate the training sample set.

Step 2: Set the number of cycles of PSO as *t*, inertia weight ω, and learning factor c_1 and c_2 increases exponentially with cycle as

$$c_1 = c_2 = e^{-t} \Delta\omega \ .$$

Step 3: Define the fitness function of PSO as the F_1 test value with weighted KNN classification.

Step 4: Initialize the value of the first particle with 1/*m* and normalized random value for remaining particles.

Each particle has *m* dimensions used to represent the properties of the position in the solution space $\omega_i = (\omega_{i1}, \omega_{i2}, ..., \omega_{im})$, ω_i means the i^{th} particle.

$$\sum_{j=1}^{m} \omega_{ij} = 1, (\omega_{i1}, \omega_{i2}, ..., \omega_{im} \in [0,1]) \ .$$

Step 5: Initialize the value of velocity for each particle randomly.

$V_i = (v_{i1}, v_{i2}, ..., v_{im})$, V_i denotes the velocity value of the i^{th} particle.

$$\sum_{j=1}^{m} v_{ij} = 0 (v_{i1}, v_{i2}, ..., v_{im} \in [-1,1]) \ .$$

Step 6: Calculate the fitness of each particle according to the fitness function; update the *pbest* for each particle and the *gbest* of whole swarm.

Step 7: Set the position of (*m*-1) dimensions and the speed of m dimensions for (*i*+1) particles by the formula (2) and (3), calculate the position value of the m^{th} dimensions according to the normalized constraint formula (2). If ω_{im}<0, means the i^{th} particle is illegal, which will be threw away, and rejoin the new initialized particle to replace it.

Step 8: While circulating, extract several rounds and particles randomly and set their position value as the initial state value.

Step 9: End the cycle until reach the present iterations, otherwise repeat step 6, 7. The program returns $gbest = (\omega_{b1}, \omega_{b2}, ..., \omega_{bm})$, where b is the index position of the best particle in the swarm.

Feature reduction. This paper uses PSO algorithm to seek the optimal weight, which represents the importance degree of each feature and decrease the dimension of feature space with it. The procedure is as follows:

Firstly, the weighted feature is looked as the basis for feature reduction. Set the feature weight reduction ratio as ε and remove this feature from the feature set F, which is used in the classification.

Secondly, remove the property whose feature weight value is 0 or small to remove the dimension of the feature space and improve the classification speed of KNN text classifier, while having little effect on classification precision.

Steps of NPSOKNN algorithm.

Step1: Re-vectorize the test set T and training set L.

Step2: Set the value of k.

Step3: Change the distance calculation function (1) as

$$sim(x, d_i) = \frac{\sum_{k=1}^{m} \omega_{bk}(\omega_k \times \omega_{ik})}{\sqrt{(\sum_{k=1}^{m} \omega_k^2)(\sum_{k=1}^{m} \omega_{ik}^2)}}; \quad (4)$$

$$\sum_{k=1}^{n} \omega_{bk} = 1.$$

Step 4: Find out the k samples in T, whose distance to the text x to be classified is the most similar. The x belongs to the category, which has the most samples among the k nearest neighbor samples.

Let $k_1, k_2, ..., k_k$ each separately stand for the number of samples from the k nearest neighbor sample to the text x that is to be classified, which actually belongs to the classes $c_1, c_2, ..., c_k$. Define the discriminant function of c_i as

$$d_i(x) = k_i, i = 1, 2, ..., k.$$

Discriminant rule as: i $d_m(x) = \max_{i=1,2,...,k} d_i(x)$, then $x \in C_m$.

This research employs 10-fold cross-validation method to evaluate the classification efficiency during the experimental process. Each data set is randomly divided into n equal subsets, one subset is selected as the test set and the remaining (n-1) subsets as the optimization training set of PSO to seek KNN feature normalized optimal weight. At the end of the first round, another subset is used as the test set and so on. Totally, n times are executed and the average and the variance of these n computing results are got finally.

Experiment and analysis. ***Data set and word segmentation pre-processing.*** In this paper, three data sets are used in the experiment: Fudan university corpus, Chinese classification corpus in the tourism field and Chinese classification corpus in sports field [9].

In this experiment, the ICTCLAS made by Chinese Academy of Sciences is used as the word segmentation processing and only the extracted nouns can be applied to classify. In addition, using *DF/ICF* (Word frequency inversion category) to serve as the method for feature selection. From the perspective of feature selection, this method proposes that selecting the entry with high-class information is the key to enhancing the classification capacity of the rare category. The result shows the feature selection effect of *DF/ICF* is better than that of *IG* and *DF*.

Using *TFIDF* rather than Boolean logic model as feature input, the formula of *TFIDF* is as follows

$$W(t, d_i) = tf(t, d_i) \times \log(D / (m + 1)).$$

Where $tf(t,d_i)$ is the ratio of the number of words t in text d_i and the total word number of d_i, D is the text number of corpus, and m shows the account of the texts which contain word t. To avoid zero denominator (i.e., all of the text do not contain the word), the denominator plus 1.

Experimental design. Cosine similarity measure function (4) is used to calculate the similarity distance measure.

10-fold cross-validation is equipped to evaluate the classification efficiency during the experimental process. Each data set is randomly divided into 10 equal subsets. Then one subset is selected as the test set and the rest 9 subsets serve as the optimization training set of PSO to seek KNN feature normalized optimal weight. Execute 10 times successively and compute the average and the variance of these 10 results finally.

This experiment uses JAVA language, compiler environment is JDK1.7, running environment is Windows 7, CPU for Intel Core i7 processor, and 8 G memory.

Comparison and analysis.

Comparison results of the best classification accuracy. The comparison of the best classification accuracy of KNN and PSOKNN in three different corpuses is as follows:

Table 1, 2 express the optimal tiny-average and macro-average of these two algorithms. What's more, the k of KNN is 10, k of KNNPSO is 7, and the iterations of PSO are 100.

Table 1

Tiny-average in three accuracy

	NPSOKNN	KNN
FuDan	0.9714285714285	0.8942857142857
tourism	0.9318926974664	0.8662500000000
sports	0.9124087591240	0.8386861313868

Table 2

Macro-average in three accuracy

	NPSOKNN	KNN
FuDan	0.9717685884294	0.8970123443239
tourism	0.9318926974664	0.8727309293727
sports	0.9171951432738	0.8490441832289

From these two tables, the tiny-average and macro-average of NPSOKNN are 0.9 above. In Fudan university corpus, the tiny-average and macro-average of NPSOKNN reach 0.97 so that the improved KNN algorithm achieved the best classification accuracy.

Comparison results of feature weight Normalization effect. Fig. 4 shows the overhead time and classification accuracy of normalized weighted and non-normalized weighted in Chinese classification corpus in the tourism field, where k=10, the number of iteration times is 100; the number of particle is 100.

In Fig. 4, the macro assessment index of normalized KNN classifier is improved and the optimal time is reduced. Fig. 5–6 respectively show the comparison of the tiny-average and macro-average of NPSOKNN and KNN with different k in three different corpuses.

Comparison results of feature reduction. The effect of classifier with different ratio feature reduction can be seen in Fig. 7. In the Chinese classification corpus in tourism field, the total account of features is 3269, ε is the reduction ratio which ε=0 means the features are not reduced. The tiny-averages and macro-averages of different reduction ratios with k=8 and k=10 are showed in figure. When the ratio is 20%, the classification accuracy reaches best. When the ratio is 20–30%, the accuracy does not change too much, the classification time can be reduced sharply.

Comparison results analysis. The accuracy of NPSOKNN is 7% higher than that of traditional KNN classifier. When the feature reduction ratio is 20%, the accuracy does not change much.

Using tiny-average as the optimal objective function, the result is similar to that of macro-average. Other distance criterions such as Markov distance and Euclidean distance, the results are similar to the result of Cosine distance.

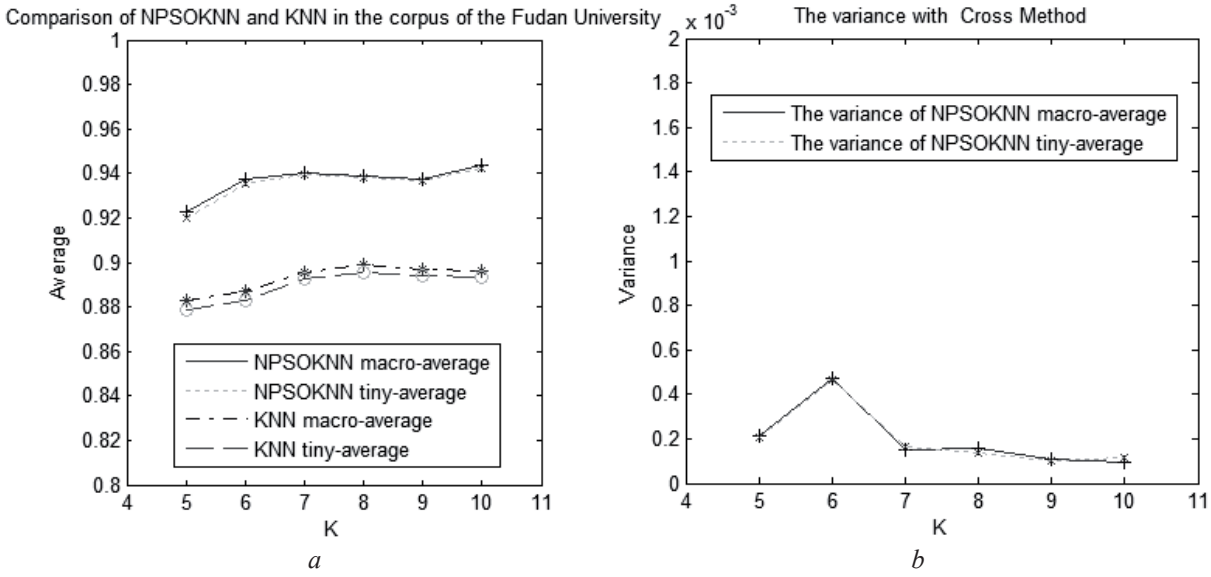


Fig. 4. Different k values in Fudan corpus: a – average k values of NPSOKNN and KNN classifier; b – variance of k values of NPSOKNN macro- and tiny-average

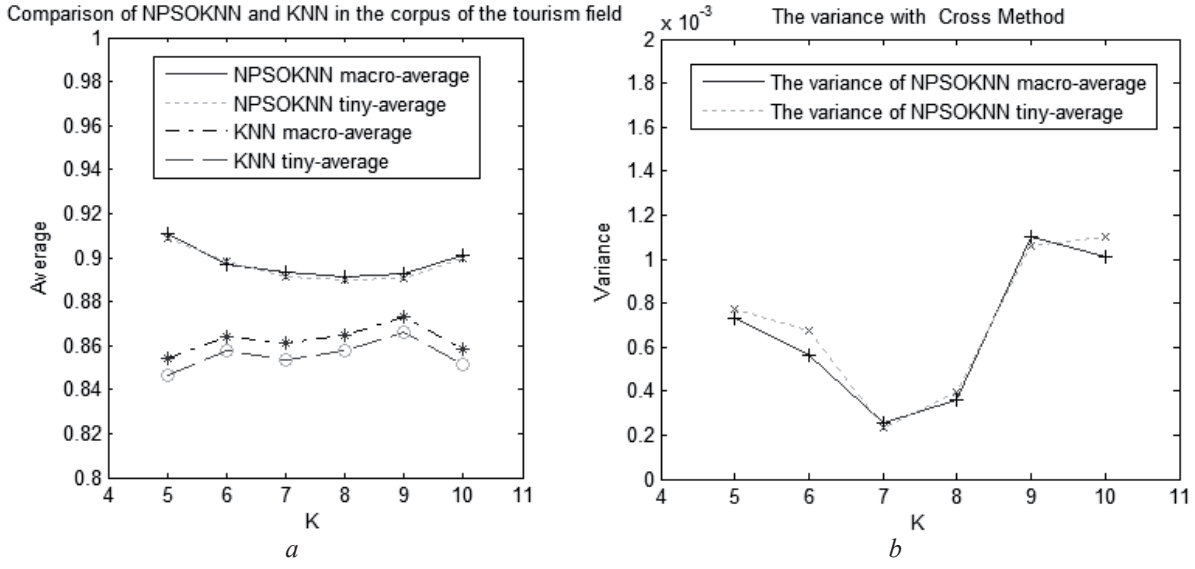
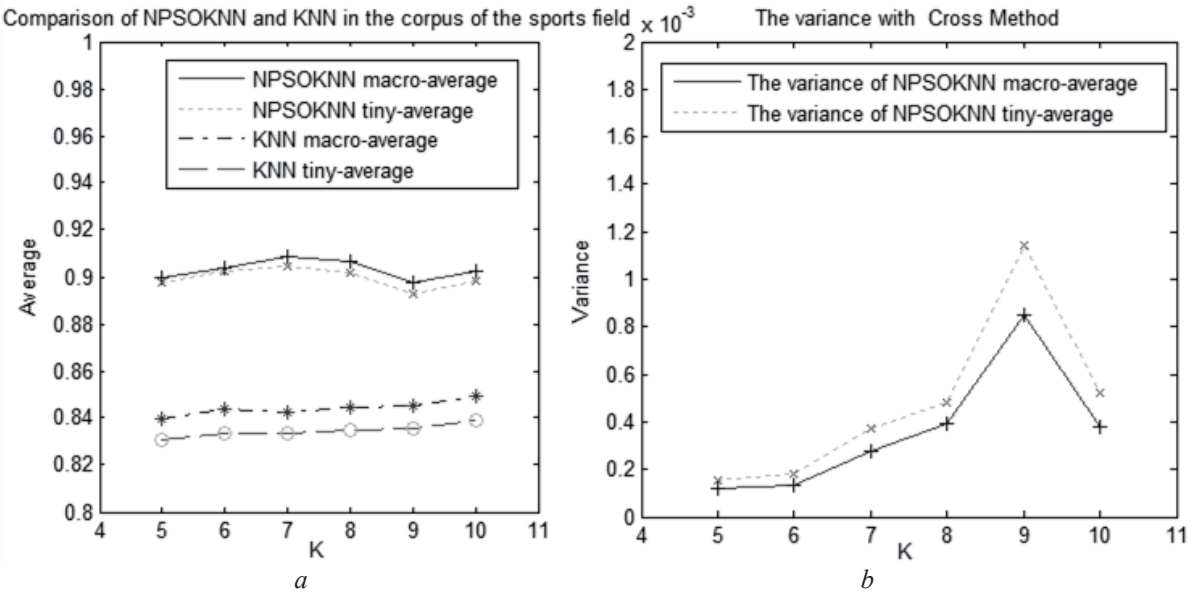
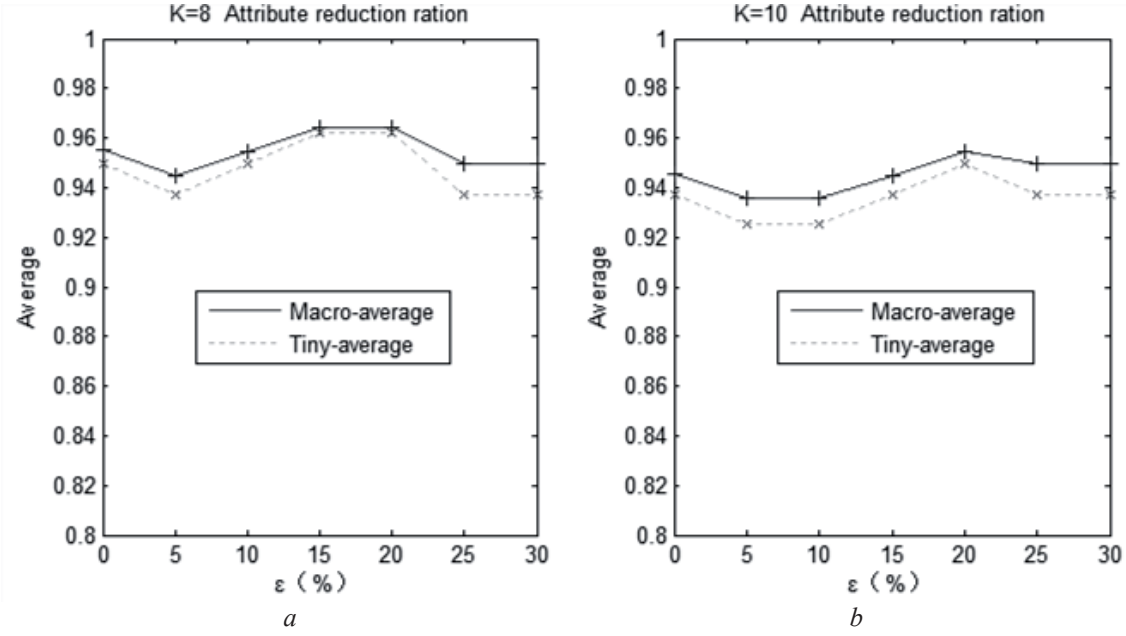


Fig. 5. Different k value in Tourism corpus: a – average k values of NPSOKNN and KNN classifier; b – variance of k values of NPSOKNN macro- and tiny-average



a *b*

Fig. 6. Different k value in Sports corpus: a –average k values of NPSOKNN and KNN classifier; b – variance of k values of NPSOKNN macro- and tiny-average



a *b*

Fig. 7. Different feature reduction ratios: a – average reduction ratio values with K=8; b − average reduction ratio values with K=10

Conclusion. A novel normalized weighted KNN text classifier is proposed in this paper. With the help of PSO, the feature weight can be self-adaptively solved and optimized. At the same time, features of KNN text classifier can be also reduced.

However, there are many parameters in KNN text classifier. We will further study the pre-self-setting of the feature weight reduction ratio ε, the selection of k, the similarity distance measure among the samples, the weighted optimization of the training set and so on.

Acknowledgements. This work was supported by National Natural Science Foundation of China (Grant No. 51305142), Natural Science Foundation of Fujian Province of China (No. 2014J01191), and project of Xiamen science and technology plan (3502Z20143041).

References / Список літератури

1. Kulkarni, S.R. and Posner, S.E., 1995. Rates of convergence of nearest neighbor estimation under arbitrary sampling. *IEEE Information Theory*, vol. 41, no. 4, pp.1028−1039.

2. Fan, J. and Lv, J.A., 2010. Selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, vol. 20, no. 1, pp. 101−148.

3. Chen, J., Huang, H., Tian, S. and Qu, Y., 2009. Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications*, vol. 36, no. 3, pp. 5432−5435.
4. Sun, W.M.B., 2012. On relationship between probabilistic rough set and Bayesian risk decision over two universes. *International Journal of General Systems*, vol. 41, no. 3, pp. 225−245.
5. Liang, J., Wang, F., Dang, C. and Qian, Y. 2014. A group incremental approach to feature selection applying rough set technique. *IEEE Knowledge and Data Engineering*, vol. 26, no. 2, pp. 294−308.
6. Chen, C.L., Tseng, F.S.C. and Liang, T., 2011. An integration of fuzzy association rules and WordNet for document clustering, *Knowledge & Information Systems*, vol. 28, no. 3, pp. 687−708.
7. Uysal, A.K., and Serkan, G., 2012. A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems*, vol. 36, pp. 226−235.
8. Confalonieri, R., Bregaglio, S. and Acutis, M. 2010. A proposal of an indicator for quantifying model robustness based on the relationship between variability of errors and of explored conditions. *Ecological Modelling*, vol. 221, no. 6, pp. 960–964.
9. Mao, Yu-Xing, Chen, Tong-Bing and Shi, Bai-Le, 2011. Efficient method for mining multiple-level and generalized association rules. *Journal of Software*, vol. 22, no. 12, pp. 2965−2980.

Мета. У стандартного класифікатора тексту по методу *k* найближчих сусідів (KNN) є ряд недоліків, таких як рівнозначність (рівноважність) усіх ознак, що знижує точність класифікації, і велика розмірність елементу, що збільшує витрати часу при обробці великих пакетів даних. Для вирішення вказаних проблем запропонований адаптивний нормалізований зважений текстовий класифікатор за методом *k* найближчих сусідів (алгоритм NPSOKNN).

Методика. Результуюча точність класифікатора використовується як цільовий показник (орієнтир) загальної оптимізації вагомості ознак. Для визначення оптимальної ваги ознак використовується оптимізація методом рою часток. Для скорочення кількості ознак і зменшення витрат часу KNN-класифікатора тексту було встановлено порогове значення, що відсікає ознаки з меншою вагою.

Результати. Проведена загальна оптимізація вагомості ознак, далі, з використанням отриманої ваги ознак і методу зменшення розмірності елементів, отриманий новий вектор ознак, розмірність якого менша, ніж у початкового за високої точності класифікації.

Наукова новизна. Проведені дослідження з удосконалення текстового класифікатора за допомогою покращених методів KNN і PSO. Розглянуті нормалізовані ваги ознак, зважені функції розрахунку відстаней, зменшення розмірності елементів. Дослідження вказаних аспектів раніше не проводилося.

Практична значимість. Результати десятиразової перехресної перевірки на достовірність показали, що середньостатистична точність алгоритму NPSOKNN вища за стандартний KNN у текстовому класифікаторові, і часові витрати істотно менші, завдяки зменшенню розмірності елементів.

Ключові слова: *класифікація тексту, метод k найближчих сусідів, нормалізована вага ознаки, оптимізація методом рою часток, зменшення розмірності елементів*

Цель. У стандартного классификатора текста по методу *k* ближайших соседей (KNN) есть ряд недостатков, таких как равнозначность (равновесность) всех признаков, которая снижает точность классификации, и большая размерность элемента, что увеличивает затраты времени при обработке больших пакетов данных. Для решения указанных проблем предложен адаптивный нормализованный взвешенный текстовый классификатор по методу *k* ближайших соседей (алгоритм NPSOKNN).

Методика. Результирующая точность классификатора используется в качестве целевого показателя (ориентира) общей оптимизации весомости признаков. Для определения оптимального веса признаков используется оптимизация методом роя частиц. Для сокращения количества признаков и уменьшения затрат времени KNN-классификатора текста было установлено пороговое значение, отсекающее признаки с меньшим весом.

Результаты. Проведена общая оптимизация весомости признаков, далее, с использованием полученных весов признаков и метода уменьшения размерности элементов, получен новый вектор признаков, размерность которого меньше, чем у исходного при высокой точности классификации.

Научная новизна. Проведены исследования по усовершенствованию текстового классификатора посредством улучшенных методов KNN и PSO. Рассмотрены нормализованные веса признаков, взвешенные функции расчета расстояний, уменьшение размерности элементов. Исследование указанных аспектов ранее не проводилось.

Практическая значимость. Результаты десятикратной перекрёстной проверки на достоверность показали, что среднестатистическая точность алгоритма NPSO KNN выше, чем у стандартного KNN в текстовом классификаторе, и временные затраты существенно меньше, благодаря уменьшению размерности элементов.

Ключевые слова: *классификация текста, метод k ближайших соседей, нормализованный вес признака, оптимизация методом роя частиц, уменьшение размерности элементов*

Рекомендовано до публікації докт. техн. наук В.В. Гнатушенком. Дата надходження рукопису 25.01.15.