Computers, Materials & Continua CMC, vol.64, no.3, pp.1741-1753, 2020

Applying Feature-Weighted Gradient Decent K-Nearest Neighbor to Select Promising Projects for Scientific Funding

Chuqing Zhang[1], Jiangyuan Yao[2, *], Guangwu Hu[3] and Thomas Schøtt[4]

Abstract: Due to its outstanding ability in processing large quantity and high-dimensional data, machine learning models have been used in many cases, such as pattern recognition, classification, spam filtering, data mining and forecasting. As an outstanding machine learning algorithm, K-Nearest Neighbor (KNN) has been widely used in different situations, yet in selecting qualified applicants for winning a funding is almost new. The major problem lies in how to accurately determine the importance of attributes. In this paper, we propose a Feature-weighted Gradient Decent K-Nearest Neighbor (FGDKNN) method to classify funding applicants in to two types: approved ones or not approved ones. The FGDKNN is based on a gradient decent learning algorithm to update weight. It updates the weight of labels by minimizing error ratio iteratively, so that the importance of attributes can be described better. We investigate the performance of FGDKNN with Beijing Innofund. The results show that FGDKNN performs about 23%, 20%, 18%, 15% better than KNN, SVM, DT and ANN, respectively. Moreover, the FGDKNN has fast convergence time under different training scales, and has good performance under different settings.

Keywords: FGDKNN, project selection, scientific funding, machine learning.

1 Introduction

In order to reduce small and medium-sized enterprises' (SMEs) financial burden and supporting their innovative activities, government agencies usually set up funds for these enterprises. As a main way for disposing scarce financial resources, pre-evaluation of application proposals is always needed. To select high-quality proposals, the funding agencies usually invite experienced employees from different industries to jointly evaluate applicants [Olbrecht and Bornmann (2010)]. The approved ones can get the fund, others cannot.

Recently, there is a growing trend for enterprises to apply for financial subsidies. However, such large number of applicants puts great pressure on project classification process due to

[1] School of Economics and Management, North China Electric Power University, Beijing, 102206, China.

[2] School of Computer Science & Cyberspace Security, Hainan University, Haikou, 570228, China.

[3] School of Computer Science, Shenzhen Institute of Information Technology, Shenzhen, 518172, China.

[4] The Faculty of Business and Social Science, University of Southern Denmark, Kolding, DK-6000, Denmark.

[*] Corresponding Author: Jiangyuan Yao. Email: yaojy@hainanu.edu.cn.

Received: 25 February 2020; Accepted: 18 April 2020.

the shortages of qualified evaluators. In order to ease the pressure, it is necessary to introduce some cutting-edge technologies to help select projects and improve efficiency.

In the past decades, various analytical methods and technologies are developed to support project selection, such as linear regression [Criscuolo, Dahlander, Grohsjean et al. (2017); Fini, Jourdan and Perkmann (2018); Teplitskiy, Acuna, Elamrani-Raoult et al. (2018)], AHP [Huang, Chu and Chiang (2008)], DEA [Eilat, Golany and Shtub (2006)], Decision Support System [Hirzel, Hettesheimer, Viebahn et al. (2018)], Markov model [Zhao, Chi and Heuvel (2015)], evidential reasoning method [Liu, Chen, Yang et al. (2018)]. However, these aforementioned methods are unable to cope highly complex data, let alone simulate human learning behavior and deal with subjective judgment uncertainties. Therefore, how to find out the evaluators' decision-making pattern and how to improve the classification accuracy are the main concerns of both funding agencies and academic scholars [Li (2017)].

Machine learning is one of the most promising technologies in classification [Hossain, Morooka, Okuno et al. (2019); Niu and Huang (2019)]. In essence, machine learning is a model aiming to find the unknown function, dependency or structure between inputs and outputs. Normally, these relations are impossible to be presented by explicit algorithms through an automatic learning process [Voyant, Notton, Kalogirou et al. (2017); Zhang, Geng, Li et al. (2019)]. Classification is a supervised learning process. It uses a set of samples of given categories to guide the classification process for the unknown category samples.

As an outstanding algorithm in statistical-based pattern recognition, KNN has achieved high classification accuracy and recall in various scenes and all types of data [Peng, Chen, Chen et al. (2018); Pan, Pan and Liu (2019)]. KNN algorithm consists of two steps: firstly, finds a group of k objects in the training set that are closest to the test object. Secondly, bases the assignment of a label on the predominance of a particular class in this neighborhood [Wu, Kumar, Quinlan et al. (2008)]. Also, attributes have different importance, while traditional KNN cannot describe them well. Facing these challenges, we propose a Feature-weighted Gradient Decent *K*-Nearest Neighbor algorithm (FGDKNN), which can be used to predict project categorization results. We use 1606 items of Beijing Innofund to investigate the performance of FGDKNN. The results show that FGDKNN outperforms traditional non-weighted KNN model significantly. The main contributions can be concluded as follows:

- FGDKNN outperforms the traditional machine technologies significantly. It performs about 23%, 20%, 18%, 15% better than KNN, SVM, DT and ANN, respectively.
- FGDKNN has good performance under different settings. Specially, it is robust with different *k* values and initial weights.
- FGDKNN has fast convergence time. For different number of training scale, rounding about 14 times can achieve more than 90% performance.

This paper is organized as follows. Section 2 introduces theoretical knowledge on machine learning methodology, including related work of project selection, KNN, and studies of feature-weighted KNN. Section 3 is a detailed explanation about the feature-weighted gradient decent *k*-nearest neighbor algorithm. Section 4 is an introduction of Beijing Innofund and data preparation. Experiment results and analysis are presented in Section 5.

Finally, concludes the paper in Section 6.

2 Related works

In this section, we provide a comprehensive literature review of project selection classifiers, including KNN and feature-weighted *k*-nearest neighbor model.

2.1 KNN classification model

KNN calculates the similarity between a target object and the most similar *k*-nearest neighbors by Euclidean distance:

$$\mathrm{d}(x, x_i) = \sqrt{\sum_{i=1}^{n}(x - x_i)^2} \tag{1}$$

where x is the target object and x_i is the *i*-th similar nearest neighbors. According to the majority vote of its neighbors, the target will be assigned to the most common class among its *k*-nearest neighbors. The classification $\delta(x_i, c_i)$ for x_i with respect to class c_i can be expressed as:

$$\delta(x_i, c_i) = \begin{cases} 1, x_i \in c_i \\ 0, x_i \notin c_i \end{cases} \tag{2}$$

This rule decides which point is nearest according to some pre-specified distance. During this procedure, all points within the neighborhood contribute equally to the final decision for x. In another word, the *k* neighbors are implicitly assumed to have an equal weight in decision, regardless of their distances to the pattern x to be classified [Zeng, Yang and Zhao (2009)].

Suárez Sánchez et al. [Suárez Sánchez, Iglesias-Rodríguez, Fernández et al. (2016)] applies KNN to predict work-related musculoskeletal disorders, and its accuracy rate is around 87%. By using KNN to predict the chances of getting diabetes, Nilashi et al. [Nilashi, Ahmadi, Shahmoradi et al. (2017)] get a more than 90% accuracy rate.

2.2 Comparison of machine learning algorithms

Decision tree follows the principle of attribute's information gain maximization to select features. SVM uses kernel function to achieve a maximum predetermined deviation. ANN is connected by artificial neurons, which is a radial basis function network based on mathematical statistics. Tab. 1 compares these models in general.

Table 1: Comparison of machine learning models

Models	Advantage	Disadvantage
KNN	The calculation process is simple and efficient	Easily influenced by the sensitivity of k and sensitive to small sample size [Li (2019)]
Decision tree	Scale well to big data, and high predictive performance for a relatively small computational effort [Rokach (2006)]	Instability and overfitting problem [Rokach (2006)]
SVM	High accuracy and good generalization for a small and unevenly distributed data [Lin (2017)]	Difficult to train large-scale data

ANN	Able to handle complex nonlinear relationships	Difficult to generalize the results due to overfitting, and lack of explanatory power [Kim and Sohn (2010)]

2.2 Feature weighted k-nearest neighbor algorithm

Besides feature selection, contributions of features to classification are also inconsistent. The distance between neighbors is determined by all the features of a sample according to the same metric. Without considering the importance of attributes, it will lead to data distortion [Wang, Zhang, Shi et al. (2019)]. Since KNN method uses system majority vote, the sensitivity of neighborhood size k always seriously degrades the KNN-based classification performance, especially in the case of small sample size with existing outliers [Huang, Lin, Huang et al. (2017)]. To solve these problems, scholars have devoted many efforts on KNN improvement.

Tan [Tan (2005)] proposes a neighbor-weighted k-nearest neighbor to solve the problem of uneven distribution. Mitani et al. [Mitani and Hamamoto (2006)] propose a local mean-based k-nearest neighbor classifier (LMKNN) by calculating the distance between each query sample and the local mean vector of k-nearest neighbors. Zeng et al. [Zeng, Yang and Zhao (2009)] design a pseudo nearest neighbor to weight distances between each query sample and k-nearest neighbors from each class. Chen et al. [Chen, Huang, Yu et al. (2013)] propose a Fuzzy KNN in improving the prediction accuracy of disease diagnose. Kuhkan [Kuhkan (2016)] allocates weight to each characteristic with different levels of importance. Huang et al. [Huang, Lin, Huang et al. (2017)] improve the KNN method to better serve the precipitation forecast scenario. Gou et al. [Gou, Yi, Du et al. (2012)] develop a local mean-based k-nearest centroid neighbor classifier to solve the problem of small sample size deficiency. And in the work of Gou et al. [Gou, Ma, Ou et al. (2019)] they propose a generalized mean distance-based k-nearest neighbor classifier (GMDKNN) by introducing multi-generalized mean distances and the nested generalized mean distance. Tab. 2 presents the comparisons of KNN classifiers mentioned above.

Table 2: Related work of KNN classifiers

Method	Target aims	Approaches	Shortcomings
Neighbor-weighted KNN [Tan (2005)]	Unbalanced data	TFIDF algorithm	Doesn't work well in large sample size
Local mean-based KNN [Mitani and Hamamoto (2006)]	Small-sample	Local mean vectors	Doesn't work well in large training sample size and mixture model data situations
Pseudo- NNR [Zeng, Yang and Zhao (2009)]	Small-sample	Distance weighted local learning	Doesn't work well in a small training sample size and singular model data case
Local mean-based k-nearest centroid neighbor classifier [Gou, Yi, Du et al. (2012)]	Small-sample	Nearest local centroid mean vector	Doesn't work well in large sample size
Fuzzy KNN [Chen, Huang, Yu et al. (2013)]	Improve accuracy	Fuzzy logic	Doesn't work well in unbalanced data
Distance weighted KNN [Kuhkan (2016)]	Sensitivity of K	Fixed distance weighted	Doesn't work well in unbalanced data
Improved KNN [Huang, Lin, Huang et al. (2017)]	Unbalanced data	Fixed distance weighted	Doesn't work well in large sample size

Generalized mean distance-based KNN (GMDKNN) [Gou, Ma, Ou et al. (2019)]	Sensitivity of K	multi-generalized mean distances	Lack of robustness and stability

3 Feature-weighted gradient decent k-nearest neighbor algorithm

The performance of feature KNN highly depends on weight settings. Intuitively, weight could be one of the characters (e.g., local mean, generalized mean). However, we find this mechanical method is insufficient for the funding allocation case. These methods mentioned above either have unsatisfactory performance or are overly complicated in operation. They can hardly be applied into the practice directly. We propose the feature-weighted gradient decent *k*-nearest neighbor algorithm (FGDKNN) to categorize projects for winning a funding or not. FGDKNN adopts a gradient decent learning based weight, which can capture the importance of characters better and classify the data set accurately. This section includes two parts, the first subsection introduces the parameters and the second shows the details of FGDKNN.

3.1 Preliminaries and notations

The objects that we are interested in can be denoted as a m-dimension vector space E. A training set T that contains n elements. T can be denoted as $\{x_1, x_2, \dots, x_n\}$, where $x_i \in E, 1 \le i \le n$. The index of the element in T can be presented as $index(x_i) = i$. The class of the element is an integer that can be presented as $class(x_i)$. The sets $T_c = \{x \in T | \ class(x) = c\}$ and $\overline{T_c} = \{x \in T | \ class(x) \neq c\}$ denote the prototypes of class c or those of a class different from c, respectively. A dissimilarity is a function that denotes as d. For two arbitrary elements y and y' in T, the distance between the two elements can be presented as $d(y, y')$. The element $x \in T$ is a *d-Nearest Neighbor(d-NN)* of element $y \in T$, when the distance between them $d(x, y) \le d(y', y), \forall\, y' \in T$. Let x be a prototype of class c and ${T'}_c = T_c - \{x\}$, which denotes the remaining element sets of T_c. The $d - NN_{{T'}_c}$ and $d - NN_{\overline{T_c}}$ are presented as $x^{=}$ and $x^{\neq}$, which means the nearnest neighbor of x in ${T'}_c$ and $\overline{T_c}$, respectively.

We now define the weighted distance between the element y and $x \in T$:

$$d(y, x) = \sqrt{\sum_{j=1}^{m} w_j^2 \left(y_j - x_j\right)^2} \tag{3}$$

where $w_j (0 \le j \le m)$ is the j-th component of x. In reality, how to set the weight could have significant influence on the model accuracy and could derive different distance mechanisms. For example, when $w_j = 1$, this corresponds to Euclidean distance. Next, we will propose a weight setting method, named feature-weighted gradient decent *k*-nearest neighbor algorithm (FGDKNN).

3.2 Learning the weight with gradient decent algorithm

The key idea of FGDKNN is to minimize the error ratio. Firstly, we define the error ratio under weight vector W as follows:

$$R_T(\boldsymbol{W}) = \frac{1}{n} \sum_{x \in T} r\left(\frac{d(x, x^{=})}{d(x, x^{\neq})}\right) \tag{4}$$

where $x^{=}$ and $x^{\neq}$are the same-class and different-class NNs of x, as defined previous. The distance function $f(.)$ could be computed according to Eq. (3). $r(.)$ is a function whose value is 0 or 1:

$$r(z) = \begin{cases} 0, if\ z < 1 \\ 1, if\ z \geq 1 \end{cases} \tag{5}$$

$R_T(\boldsymbol{W})$ can be regarded as the estimate of the misclassification probability over the training set T, since $d(x, x^{=}) > d(x, x^{\neq})$ denotes the distance between the NN of the same class is larger. It might be an error case. Otherwise, the distance between x and the element of different class is larger, which respects the real condition. As $r(.)$ is not smoothly, some approximations are needed. We use the sigmoid as the approximate function:

$$R_T(\boldsymbol{W}) \approx \frac{1}{n}\left(\sum_{z \in T} G_\beta(r(z))\right) \tag{6}$$

where $G_\beta(.)$ is the sigmoid function with slope β, centralized at α:

$$G_\beta(\alpha) = \frac{1}{1+e^{\beta(1-\alpha)}} \tag{7}$$

Clearly, when β is large enough, this approximation is very accurate. On the other hand, if it is small, the contribution of each NN classification error to the index R_T is more important depending on the corresponding quotient of the distance responsible error.

The key idea of FGDKNN is to train the feature weight vector (i.e., $\boldsymbol{W}$) according to the estimate of the misclassification probability (i.e., $R_T(\boldsymbol{W})$). To minimize $R_T(\boldsymbol{W})$, we derive a step based weight update method by changing w a small amount, μ_j, in the negative direction of the gradient of $R_T(\boldsymbol{W})$. Assume there are S steps in total, at each step $s \leq S$, the weight w_j^s is updated as follows:

$$w_j^{s+1} = w_j^s - \mu_j \left(\frac{\partial R_T(\boldsymbol{W})}{\partial w_j}\right)^s \tag{8}$$

μ_j is the learning rates. In reality, its value could be fixed and is inversely proportional to the variance of each feature. According to Eqs. (6)-(8), we have:

$$\frac{\partial R_T(\boldsymbol{W})}{\partial w_j} \approx \frac{1}{n}\sum_{\forall\, x \in T} G'_\beta\left(r(x)\right) \frac{\left(x_j - x_j^{=}\right)^2}{d(x,x^{=})d(x,x^{\neq})} w_j - \frac{1}{n}\sum_{\forall\, x \in T} G'_\beta\left(r(x)\right) \frac{\left(x_j - x_j^{\neq}\right)^2 d(x,x^{=})}{d^3\ (x,x^{\neq})} w_j \tag{9}$$

where $G'_\beta\left(r(x)\right)$ is the derivatives of $G_\beta(\alpha)$ which can be computed as:

$$G'_\beta(\alpha) = \frac{dG_\beta(\alpha)}{d\alpha} = \frac{\beta e^{\beta(1-\alpha)}}{\left(1+e^{\beta(1-\alpha)}\right)^2} \tag{10}$$

Derivative of $S_\beta(\alpha)$ could be maximal when $\alpha = 1$. When β is large, $\frac{dG_\beta(\alpha)}{d\alpha}$ would approach Dirac delta function; conversely, it is approximately constant for a wide range of values of α.

With Eqs. (9) and (10), we now derive the weight update rule as:

$$w_j^{s+1} = w_j^s - \mu_j w_j^s \left(\frac{1}{n}\sum_{\forall\, x \in T} G'_\beta\left(r(x)\right) \frac{\left(x_j - x_j^{=}\right)^2}{d(x,x^{=})d(x,x^{\neq})} - \frac{1}{n}\sum_{\forall\, x \in T} G'_\beta\left(r(x)\right) \frac{\left(x_j - x_j^{\neq}\right)^2 d(x,x^{=})}{d^3\ (x,x^{\neq})}\right) \tag{11}$$

In each step s, the weight vector updates according to Eq. (11). Until the update round number reaches S, we can attain the final weight vector.

The effects of Eqs. (8)-(11) is clear, in each step, the weight is updated associated with the error. If there are too much errors, the weight changes to the opposite direction. After enough iterations, the weight can capture the importance of labels better.

4 Materials and methods

In this section, we first introduce project selection process of Beijing Innofund, and then introduce the data pre-processing methods.

4.1 Project selection of Beijing Innofund

Small and medium-sized technology-based enterprise special fund in Beijing (known as "Beijing Innofund") is initiated by Beijing municipal government. It is a public funding aiming to support SMEs' technological innovation activities and foster their growth. During its ten years' development, Beijing Innofund has achieved a great accomplishment. It has spent over 1.34 billion RMB to support over 3000 technology-based small and medium-sized enterprises.

The funding rate of Beijing Innodunf is less than 25%. We got 1633 applicants for Beijing Innofund of year 2017 from its funding agency-Beijing municipal science & technology commission. And those winning the funding is around 400. We will use these data to test the classification accuracy of FGDKNN.

As the predicting items of Beijing Innofund are of different units, normalization is needed. In this study, we scaled the data into the interval of [0, 1] by (12):

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{12}$$

where x' is the original value, x' is the scaled value, $\max(x)$ is the maximum value of feature x, and $\min(x)$ is the minimum value of feature x.

4.2 Feature selection with logistic regression

The data given for classification should not contain irrelevant or redundant attributes, otherwise it will increase the processing time and might degrade the quality of discovered patterns. Due to the diversity and complexity of the evaluation indicators, the factors that contribute to the evaluation are often submerged among many interfering factors. Those factors that are not meaningful to the evaluation results may overshadow the information presented, making it more difficult to discover meaningful patterns from the data, thus affecting the accuracy of classification. Therefore, it is necessary to select influential factors first [Kalpana, Saravanan and Vivekanandan (2012)].

As mentioned above, the funding results of Beijing Innofund contain two types: approved (i.e., 1) or not (i.e., 0). Since there are only two choices, we choose Logistic Regression to derive the correlation. Let P denote the result of the Innofund, the Logistic Regression is computed as follows:

$$Logit(P) = \log\left(\frac{P}{1-P}\right) = \alpha + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = \alpha + wx \tag{13}$$

where w_1 can present the correlation between the probability results and the variable, and it can be obtained by the maximum likelihood estimate. According to the Innofund rating scale, we put all the potential relevant variables into the correlation model, and find out that only 19 features of them are of high correlations to funding decision (as shown in Tab. 3). The

coefficient between variables like asset (log), avenue (log), net asset (log), founder's education, project manager's education, the number of R & D stuff, project phase, project R&D investment (log), expected asset increase in two years (log), expected revenue increase in two years (log), the number of invention, the number of software copyright, firm R & D investment (log), angel investment, pre-A investment, institution recommendation, prize winner, firm age, firm size, founder type are significantly related to funding decision.

Table 3: Feature selection results and the corresponding importance

Variables	Explanation	Importance
Firm size	Total employees	0.012
Firm age	Established year	0.084
Field	Application field	0.031
Assets	Firm's total assets	0.098
Revenue	Firm's total revenue	0.006
Net asset	Firm's total net asset	0.307
RD stuff	Total number of R & D stuff	0.002
RD	The total amount of R & D expenditure	0.270
Angel	Get angel investment as 1; others as 0	0.410
Pre-A	Get Pre-A investment as 1; others as 0	0.693
Community support	Get community support as 1; others as 0	0.407
Prize winner	Innovative competition winner as 1; others as 0	0.436
Invention	The total amount of inventions	0.172
Copyright	The total amount of copyrights	0.087
Founder type	Student, returnee, researcher, manager	0.007
Fdedu	Founder's education level	0.358
Pmedu	Project manager's education level	0.234
Project phase	Pilot, testing, or on the market	0.301
Project RD	The total amount of project's R & D expenditure	0.001
Project asset	Project's total assets in two years	0.060
Project revenue	Project's total revenue in two years	0.046

From Tab. 3, we can see pre-A investment is the most importance variables to funding decision. Its attribute significance is near 70%. Prize winner, angel investment and community support rank after pre-A investment with significance all above 40%. Firm's net asset, R & D expenditure, firm's asset, number of invention, founder's education level, project manager's education level, project phase are influential factors to assessors' funding decision with significance importance over 10%. On the other hand, the remaining factors, like firm age, firm size, field, firm's revenue, number of R & D stuff, number of copyrights, founder type, project's R & D expenditure, project's future asset and revenue cannot influence assessors' decision much, since their significance importance all under 10%.

5 Evaluation

In this section, we evaluate the performance of FGDKNN with the Beijing Innofund. The data is divided into two parts randomly, where the training set contains 1000 items and the others are test set. Let TP, FP, FN, TN denote True Positive, False Positive, False Negative, True Negative, respectively. We use Precision, Recall and accuracy as the performance metric to test the accuracy of FGDKNN:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{14}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{15}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{16}$$

Besides the equal weight KNN, some other machine learning technologies, such as Support Vector Machine (SVM), Decision Trees (DT) and Artificial Neutral Network (ANN) are also included in our experiments. Main results of our evaluation can be concluded as:

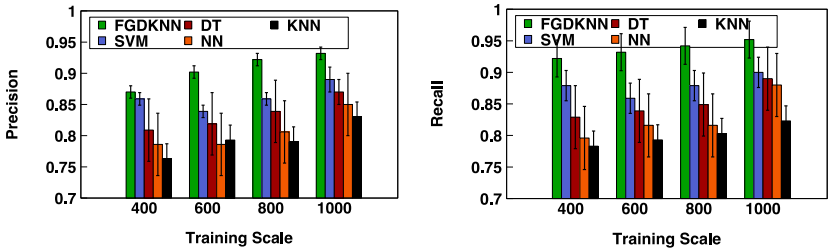
FGDKNN can outperform the traditional machine technologies significantly. For the prediction accuracy, it performs about 23%, 20%, 18%, 15% better than KNN, SVM, DT and ANN, respectively.

FGDKNN has good performance under different settings. Specially, it is robust with different *k* values and the initial weight computed by Logistic Regression is about 5%-10% better than Pearson Correlation Coefficient (PCC), Distance Correlation (DC) with different training data scale.

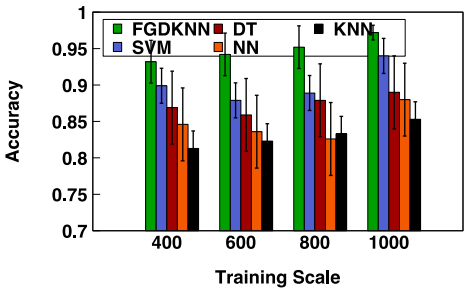
FGDKNN has fast convergence time, for different number of training set scale, rounding about 14 times can achieve more than 90% performance.

5.1 Performance comparison with different metrics

Firstly, we evaluate the performance of FGDKNN with different machine learning methods. Fig. 1 shows the comparison under different comparison metrics with different training scales. For the prediction precision, as Fig. 1(a) shown, we can see that FGDKNN performs about 19%, 15%, 13%, 10% better than KNN, SVM, DT and NN, respectively. Fig. 1(b) shows that FGDKNN performs about 25%, 24%, 22%, 18% better than KNN, SVM, DT and NN, respectively. The similar result for accuracy is shown in Fig. 1(c). On average, FGDKNN performs about 23%, 20%, 18%, 15% better than KNN, SVM, DT and NN, respectively. FGDKNN has larger prediction values on True Positive and False Negative, since its weight update rule can capture the characters of data better. Also, the results of all methods perform better with larger scale of training set.



(a) Precision (b) Recall



(c) Accuracy

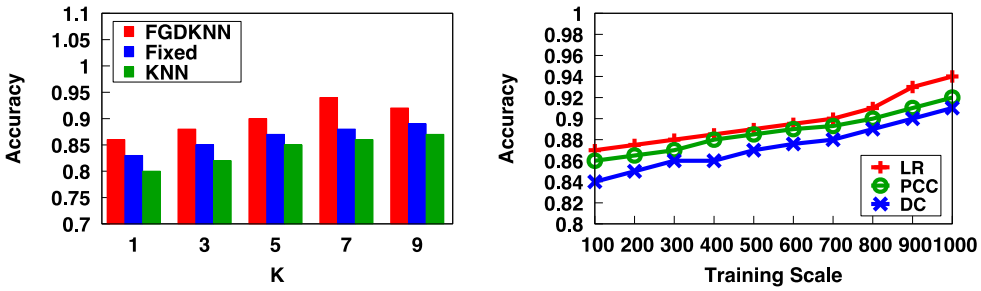
Figure 1: Performance comparison under different metrics

5.2 Performance of FGDKNN under different settings

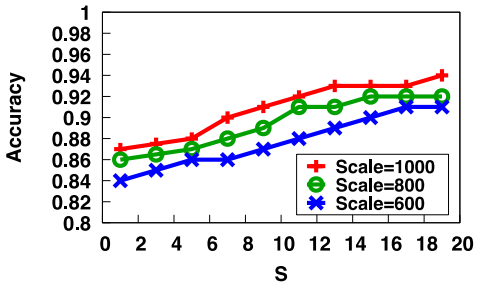
Firstly, we investigate performance with k. Fig. 2(a) shows that with the increasing of k, the accuracy of FGDKNN, Fixed Weight KNN and KNN first increase, then decrease. The accuracy of FGDKNN even reaches 94%, when k is 7. The accuracy of Fixed Weight KNN ranges from 82% to 87%. Performance of FGDKNN is about 15% better than Fixed Weight KNN on average. The overall accuracy rate of FGDKNN outperforms KNN at least 10%. That is to say, our proposed FGDKNN model gives an improvement over the traditional approach with the increasing of the neighborhood size k. It verifies that feature significance does play an important role in improving prediction accuracy.

In the previous section, we use the Logistic Regression to determine the initial value of weight vectors. Indeed, some other methods (Pearson Correlation Coefficient, Distance Correlation, etc.) can also be used to decide the importance. Fig. 2(b) shows the comparison of different initial weight settings. We can see that Logistic Regression (LR) performs about 5%-10% better than Pearson Correlation Coefficient (PCC), Distance Correlation (DC) with different training data scales. The reason for this is that Logistic Regression computes correlation with total training step.

S in the gradient decent which can decide the iteration number of weight decisions. Intuitively, the larger S could capture the characters of data well and when S is large enough, the accuracy will be stable, since the parameters can describe the characters of data well. Fig. 2(c) shows the results of this. We can see that rounding about 14 times can achieve more than 90% performance for different training scales.



(a) (b)



(c)

Figure 2: Performance under different settings

6 Discussion and conclusion

In predicting which applicants will get funded by Beijing Innofund, we adopt a gradient decent learning based weight with the flexible weight adjustment KNN method. It overcomes the traditional KNN's shortcomings with some small-sized unbalanced data distributions. We find FGDKNN can capture the characters of data well and classify the data set accurately. The FGDKNN performs better than classical non-weighted KNN by over 15%, with its satisfactory accuracy rate. The method breaks a new path for weights assigning, and we prove that FGDKNN method can be used in project evaluation field.

Acknowledgement: We would like to thank Beijing municipal government for support us doing this research.

Funding Statement: J. Yao would like to thank the support of Program of Hainan Association for Science and Technology Plans to Youth R & D Innovation [QCXM201910], Scientific Research Setup Fund of Hainan University [KYQD (ZR) 1837], the National Natural Science Foundation of China [61802092], and G. Hu would like to thank the support of Fundamental Research Project of Shenzhen Municipality [JCYJ20170817115335418].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Chen, H. L.; Huang, C. C.; Yu, X. G.; Xu, X.; Sun, X. et al. (2013): An efficient diagnosis system for detection of parkinson's disease using fuzzy k-nearest neighbor approach. *Expert Systems with Applications*, vol. 40, no. 1, pp. 263-271.

Chen, Z. Z.; Li, L.; Yao, Z. A. (2005): Feature-weighted k-nearest neighbor algorithm with svm. *Acta Scientiarum Naturalium Universitatis Sunyatseni*, vol. 44, no. 1, pp. 17-20.

Criscuolo, P.; Dahlander, L.; Grohsjean, T.; Salter, A. (2016): Evaluating novelty: the role of panels in the selection of R & D projects. *Academy of Management Journal*, vol. 60, no. 2, pp. 433-460.

Eilat, H.; Golany, B.; Shtub, A. (2006): R & D project evaluation: an integrated DEA and balanced scorecard approach. *Omega*, vol. 36, no. 5, pp. 895-912.

Fini, R.; Jourdan, J.; Perkmann, M. (2018): Social valuation across multiple audiences: the interplay of ability and identity judgements. *Academy of Management Journal*, vol. 61, no. 6, pp. 2230-2264.

Gou, J.; Ma, H.; Ou, W.; Zeng, S.; Rao, Y. et al. (2019): A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications*, vol. 115, pp. 356-372.

Gou, J.; Yi, Z.; Du, L.; Xiong, T. (2012): A local mean-based k-nearest centroid neighbor classifier. *The Computer Journal*, vol. 55, no. 9, pp. 1058-1071.

Hirzel, S.; Hettesheimer, T.; Viebahn, P.; Fischedick, M. (2018): A decision support system for public funding of experimental development in energy research. *Energies*, vol. 11, no. 6, pp. 1357.

Hossain, B.; Morooka, T.; Okuno, M.; Nii, M.; Yoshiya, S. (2019): Surgical outcome prediction in total knee arthroplasty using machine learning. *Intelligent Automation and Soft Computing*, vol. 25, no. 1, pp. 105-115.

Huang, C. C.; Chu, P. Y.; Chiang, Y. H. (2008): A fuzzy AHP application in government-sponsored R & D project selection. *Omega*, vol. 36, no. 6, pp. 1038-1052.

Huang, M.; Lin, R.; Huang, S.; Xing, T. (2017): A novel approach for precipitation forecast via improved k-nearest neighbor algorithm. *Advanced Engineering Informatics*, vol. 33, pp. 89-95.

Kalpana, B.; Saravanan, V.; Vivekanandan, K. (2012): A survey of feature selection models for classification. *International Journal of Advanced Research in Computer Science*, vol. 3, no. 1, pp. 131-136.

Kim, H., Sohn, S. (2010): Support vector machines for default prediction of SMEs based on technology credit. *European Journal of Operational Research*, vol. 201, no. 3, pp. 838-846.

Kuhkan, M. (2016): A method to improve the accuracy of k-nearest neighbor algorithm. *International Journal of Computer Engineering and Information Technology*, vol. 8, no. 6, pp. 90-95.

Li, D. (2017): Expertise versus bias in evaluation: evidence from the NIH. *American Economic Journal Applied Economics*, vol. 9, no. 2, pp. 60-92.

Li, J. (2019): An improved K-nearest neighbor algorithm using tree structure and pruning technology. *Intelligent Automation and Soft Computing*, vol. 25, no. 1, pp. 35-48.

Lin, S. (2017): Integrated artificial intelligence-based resizing strategy and multiple criteria decision making technique to form a management decision in an imbalanced environment. *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 6, pp. 1981-1992.

Liu, F.; Chen, Y. W.; Yang, J. B.; Xu, D. L.; Liu, W. S. (2018): Solving multiple-criteria R & D project selection problems with a data-driven evidential reasoning rule. *International Journal of Project Management*, vol. 37, no. 1, pp. 87-97.

Mitani, Y.; Hamamoto, Y. (2006): A local mean-based nonparametric classifier. *Pattern Recognition Letters*, vol. 27, no. 10, pp. 1151-1159.

Nilashi, M.; Ahmadi, H.; Shahmoradi, L.; Mardani, A.; Ibrahim, O. et al. (2017): Knowledge discovery and diseases prediction: a comparative study of machine learning techniques. *Journal of Soft Computing and Decision Support Systems*, vol. 4, no. 5, pp. 8-16.

Niu, B.; Huang, Y. (2019): An improved method for web text affective cognition computing based on knowledge graph. *Computers, Materials & Continua*, vol. 59, no. 1, pp. 1-14.

Olbrecht, M.; Bornmann, L. (2010): Panel peer review of grant applications: what do we know from research in social psychology on judgment and decision-making in groups? *Research Evaluation*, vol. 19, no. 4, pp. 293-304.

Pan, N.; Pan, D.; Liu, Y. (2019): The crime scene tools identification algorithm based on GVF-Harris-SIFT and KNN. *Intelligent Automation and Soft Computing*, vol. 25, no. 2, pp. 413-419.

Peng, Y.; Chen, D.; Chen, L.; Yu, J.; Bao, M. (2018): The machine learning based finite element analysis on road engineering of built-in carbon fiber heating wire. *Intelligent Automation and Soft Computing*, vol. 24, no. 3, pp. 531-539.

Suárez Sánchez, A.; Iglesias-Rodríguez, F. J.; Riesgo Fernández, P.; de Cos-Jiez, F. J. (2016): Applying the k-nearest neighbor technique to the classification of workers according to their risk of suffering musculoskeletal disorders. *International Journal of Industrial Ergonomics*, vol. 52, no. 9, pp. 92-99.

Tan, S. (2005): Neighbor-weighted k-nearest neighbor for unbalanced text corpus. *Expert Systems with Applications*, vol. 28, no. 4, pp. 667-671.

Teplitskiy, M.; Acuna, D.; Elamrani-Raoult, A.; Körding, K.; Evans, J. (2018): The sociology of scientific validity: how professional networks shape judgement in peer review. *Research Policy*, vol. 47, no. 9, pp. 1825-1841.

Voyant, C.; Notton, G.; Kalogirou, S.; Nivet, M. L.; Paoli, C. et al. (2017): Machine learning methods for solar radiation forecasting: a review. *Renewable Energy*, vol. 105, pp. 569-582.

Wang, Z.; Zhang, H.; Shi, X.; Yin, X.; Li, Y. et al. (2019): Efficient scheduling of weighted coflows in data centers. *IEEE Transactions on Parallel Distributed Systems*, vol. 30, no. 9, pp. 2003-2017.

Wu, X.; Kumar, V.; Quinlan, J. R.; Ghosh, J.; Yang, Q. et al. (2008): Top 10 algorithms in data mining. *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1-37.

Zeng, Y.; Yang, Y.; Zhao, L. (2009): Pseudo nearest neighbor rule for pattern classification. *Expert Systems with Applications*, vol. 36, no. 2, pp. 3587-3595.

Zhang, H.; Geng, H.; Li, Y.; Yin, X.; Shi, X. et al. (2019): DA & FD-deadline-aware and flow duration-based rate control for mixed flows in DCNs. *IEEE/ACM Transactions on Networking*, vol. 27, no. 6. pp. 2458-2471.

Zhao, Y. W.; Chi, C. H.; Heuvel, W. J. (2015): Imperfect referees: reducing the impact of multiple biases in peer review. *Journal of the Association for Information Science and Technology*, vol. 66, no. 11, pp. 2340-2356.