International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

COPY RIGHT

ELSEVIER
SSRN

IJIEMR Transactions, online available on 24th Feb 2018. Link
:http://www.ijiemr.org/downloads.php?vol=Volume-08&issue=ISSUE-02

Title: A NEW APPROACH FOR KNN CLASSIFICATION BY USING DIFFERENT NUMBER OF NEAREST NEIGHBOUR

Volume 08, Issue 02, Pages: 78–84.

Paper Authors

MR.K.SRINIVASA RAO, B.ANIL KUMAR
Vignan's Lara Institute of Technology & Science

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

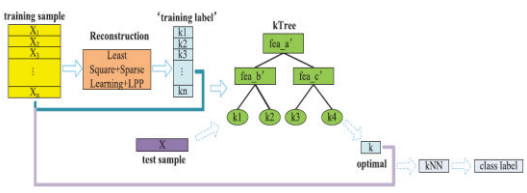
International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

A NEW APPROACH FOR KNN CLASSIFICATION BY USING DIFFERENT NUMBER OF NEAREST NEIGHBOUR

MR.K.SRINIVASA RAO[1], B.ANIL KUMAR[2]
Assistant Professor[1], Department of M.C.A ,Vignan's Lara Institute of Technology & Science
M.C.A Student[2],Department of M.C.A ,Vignan's Lara Institute of Technology & Science

Abstract:
K nearest neighbor (kNN) method is a popular classification method in data mining and statistics because of its simple implementation and significant classification performance. However, it is impractical for traditional kNN methods to assign a fixed k value (even though set by experts) to all test samples. Previous solutions assign different k values to different test samples by the cross validation method but are usually timeconsuming. This paper proposes a kTree method to learn different optimal k values for different test/new samples, by involving a training stage in the kNN classification. Specifically, in the training stage, kTree method first learns optimal k values for all training samples by a new sparse reconstruction model, and then constructs a decision tree (namely, kTree) using training samples and the learned optimal k values. In the test stage, the kTree fast outputs the optimal k value for each test sample, and then, the kNN classification can be conducted using the learned optimal k value and all training samples. As a result, the proposed kTree method has a similar running cost but higher classification accuracy, compared with traditional kNN methods, which assign a fixed k value to all test samples. Moreover, the proposed kTree method needs less running cost but achieves similar classification accuracy, compared with the newly kNN methods, which assign different k values to different test samples. This paper further proposes an improvement version of kTree method (namely, k*Tree method) to speed its test stage by extra storing the information of the training samples in the leaf nodes of kTree, such as the training samples located in the leaf nodes, their kNNs, and the nearest neighbor of these kNNs. We call the resulting decision tree as k*Tree, which enables to conduct kNN classification using a subset of the training samples in the leaf nodes rather than all training samples used in the newly kNN methods. This actually reduces running cost of test stage. Finally, the experimental results on 20 real data sets showed that our proposed methods (i.e., kTree and k*Tree) are much more efficient than the compared methods in terms of classification tasks.

Architecture



Introduction:
DIFFERENT from model-based methods which first learn models from training samples and then predict test samples with the learned model [1]–[6], the model-free k nearest neighbors (kNNs) method does not

have training stage and conducts classification tasks by first calculating the distance between the test sample and all training samples to obtain its nearest neighbors and then conducting kNN classification1 (which assigns the test samples with labels by the majority rule on the labels of selected nearest neighbors). Because of its simple implementation and significant classification performance, kNN method is a very popular method in data mining and statistics and thus was voted as one of top ten data mining algorithms [7]–[13].

Previous kNN methods include: 1) assigning an optimal k value with a fixed expert-predefined value for all test samples [14]–[19] and 2) assigning different optimal k values for different test samples [18], [20], [21]. For example, Lall and Sharma [19] indicated that the fixed optimal-k-value for all test samples should be $k = \sqrt{n}$ (where $n > 100$ and n is the number of training samples), while Zhu et al. [21] proposed to select different optimal k values for test samples via tenfold cross validation method. However, the traditional kNN method, which assigns fixed kNNs to all test samples (fixed kNN methods for short), has been shown to be impractical in real applications. As a consequence, setting an optimal-k-value for each test sample to conduct kNN classification (varied kNN methods for short) has been becoming a very interesting research topic in data mining and machine learning [22]–[29].A lot of efforts have been focused on the varied kNN methods, which efficiently set different optimal-k-values to different samples [20], [30], [31]. For example, Li et al. [32] proposed to use different numbers of nearest neighbors for different categories and Sahigara et al. [33] proposed to employ the Monte Carlo validation method to select an optimal smoothing parameter k fore each test sample. Recently, Cheng et al. [20] proposed a sparse-based kNN method to learn an optimal-k-value for each test sample and Zhang et al. [30] studied the kNN method by learning a suitable k value for each test sample based on a reconstruction framework [34]. Previous varied kNN methods usually first learn an individual optimal-k-value for each test sample and then employ the traditional kNN classification (i.e., majority rule on k training samples) to predict test samples by the learned optimal-k-value. However, either the process of learning an optimal-k-value for each test sample or the process of scanning all training samples for finding nearest neighbors of each test sample is time-consuming. Therefore, it is challenging for simultaneously addressing these issues of kNN method, i.e., optimal-k-values learning for different samples, time cost reduction, and performance improvement.

To address aforementioned issues of kNN methods, in this paper, we first propose a kTree2 method for fast learning an optimal-k-value for each test sample, by adding a training stage into the traditional kNN method and thus outputting a training model, i.e., building a decision tree (namely, kTree) to predict the optimal-k-values for all test samples. Specifically, in the training

stage, we first propose to reconstruct each training sample by all training samples via designing a sparse-based reconstruction model, which outputs an optimal-k-value for each training sample. We then construct a decision tree using training samples and their corresponding optimal-k-values, i.e., regarding the learned optimal-k-value of each training sample as the label. The training stage is offline and each leaf node stores an optimal-k-value in the constructed kTree. In the test stage, given a test sample, we first search for the constructed kTree (i.e., the learning model) from the root node to a leaf node, whose optimal-k-value is assigned to this test sample so that using traditional kNN classification to assign it with a label by the majority rule.

There are two distinguished differences between the previous kNN methods [20], [30] and our proposed kTree method. First, the previous kNN methods (e.g., fixed kNN methods and varied kNN methods) have no training stage, while our kTree method has a sparse-based training stage, whose time complexity is O(n2) (where n is the sample size). It is noteworthy that the training stage of our kTree method is offline. Second, even though both the varied kNN methods and our proposed kTree method (which can be regarded as one of varied kNN methods) first search the optimal-k-values and then conduct traditional kNN classification to classify the test sample with the learned optimal-k-values, the previous methods need at least O(n2) time complexity to obtain the optimal-k-values due to involving a sparse-based learning process, while our kTree method only needs O(log(d) + n) (where d is the dimensions of the features) to do that via the learned model, i.e., the kTree. It is also noteworthy that the process of traditional fixed kNN method assigning a fixed k value to all test samples needs at least O(n2d) via scanning all training samples for each test sample.

Existing system:

There are two distinguished differences between the previous kNN methods [20], [30] and our proposed kTree method. First, the previous kNN methods (e.g., fixed kNN methods and varied kNN methods) have no training stage, while our kTree method has a sparse-based training stage, whose time complexity is O(n2) (where n is the sample size). It is noteworthy that the training stage of our kTree method is offline. Second, even though both the varied kNN methods and our proposed kTree method (which can be regarded as one of varied kNN methods) first search the optimal-k-values and then conduct traditional kNN classification to classify the test sample with the learned optimal-k-values, the previous methods need at least O(n2) time complexity to obtain the optimal-k-values due to involving a sparse-based learning process, while our kTree method only needs O(log(d) + n) (where d is the dimensions of the features) to do that via the learned model, i.e., the kTree. It is also noteworthy that the process of traditional fixed kNN method assigning a fixed k value to all test samples needs at least O(n2d) via scanning all training samples for each test sample.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Proposed system:

In this paper, we first propose a kTree2 method for fast learning an optimal-k-value for each test sample, by adding a training stage into the traditional kNN method and thus outputting a training model, i.e., building a decision tree (namely, kTree) to predict the optimal-k-values for all test samples. Specifically, in the training stage, we first propose to reconstruct each training sample by all training samples via designing a sparse-based reconstruction model, which outputs an optimal-k-value for each training sample. We then construct a decision tree using training samples and their corresponding optimal-k-values, i.e., regarding the learned optimal-k-value of each training sample as the label. The training stage is offline and each leaf node stores an optimal-k-value in the constructed kTree. In the test stage, given a test sample, we first search for the constructed kTree (i.e., the learning model) from the root node to a leaf node, whose optimal-k-value is assigned to this test sample so that using traditional kNN classification to assign it with a label by the majority rule.

Modules:

1. **Reconstruction**

   Denote by training samples represent the number of trainingsamples and features, in this section, we design to use trainingsamplesto reconstruct themselves, i.e., reconstruct eachtraining sample, with the goal that the distance betweenand(where the reconstructioncoefficient matrix) is as small as possible. To do this, we usea least square loss function.

2. **K Tree Method**

   The kNN based on graph sparse reconstruction (GS-kNN) method in used to reconstruct test samples by training samples to yield good performance. However, it is time consuming predicting each test sample, where n is the number of training samples. To overcome this, we propose a training stage to construct a k-decision tree (namely, kTree) between training samples and their corresponding optimal-k-values. The motivation of our method is that we expect to find the relationship between training samples and their optimal-k-values so that the learned kTree enables to output an optimal-k-value for a test sample in the test stage.

3. **k*Tree Classification**

   In the training stage, the proposed k*Tree method constructs the decision tree (namely, k*Tree) by using the same steps of kTree described in Section III-D. Their difference is the information in the leaf nodes. That is, kTree stores the optimalk- value in leaf nodes, while k*Tree stores the optimal-k-value as well as other information in the leaf nodes, including a subset of training samples located in this leaf node, the kNNs of each sample in this subset, and the nearest neighbor of each of these kNNs.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

Algorithm

kNN algorithm

In pattern recognition, the ***k*-nearest neighbors algorithm** (***k*-NN**) is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

- In *k-NN classification*, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of that single nearest neighbor.

- In *k-NN regression*, the output is the property value for the object. This value is the average of the values of its *k* nearest neighbors.

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The *k*-NN algorithm is among the simplest of all machine learning algorithms.

Conclusion:

In this paper, we have proposed two new kNN classification algorithms, i.e., the kTree and the k*Tree methods, to select optimal-k-value for each test sample for efficient and effective kNN classification. The key idea of our proposed methods is to design a training stage for reducing the running cost of test stage and improving the classification performance. Two set of experiments have been conducted to evaluate the proposed methods with the competing methods, and the experimental results indicated that our methods outperformed the competing methods in terms of classification accuracy and running cost. In future, we will focus on improving the performance of the proposed methods on high-dimensional data.

References

[1] S. Zhang, "Shell-neighbor method and its application in missing data imputation," Appl. Intell., vol. 35, no. 1, pp. 123–133, 2011.

[2] L. Shao, L. Liu, and X. Li, "Feature learning for image classification via multiobjective genetic programming," IEEE Trans. Neural Netw. Learn.Syst., vol. 25, no. 7, pp. 1359–1371, Jul. 2014.

[3] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 7, pp. 1088–1099, Jul. 2006.

[4] J. Yu, X. Gao, D. Tao, X. Li, and K. Zhang, "A unified learning framework for single image super-resolution," IEEE Trans. Neural Netw. Learn. Syst., vol. 25, no. 4, pp. 780–792, Apr. 2014.

[5] Q. Zhu, L. Shao, X. Li, and L. Wang, "Targeting accurate object extraction from an image: A comprehensive study of natural image matting," IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 2, pp. 185–207, Feb. 2015.
[6] K. Huang, D. Tao, Y. Yuan, X. Li, and T. Tan, "Biologically inspired features for scene classification in video surveillance," IEEE Trans. Syst., Man, Cybern., Part B, vol. 41, no. 1, pp. 307–313, Feb. 2011.
[7] S. Zhang, "Nearest neighbor selection for iteratively KNN imputation," J. Syst. Softw., vol. 85, no. 11, pp. 2541–2552, 2012.
[8] T. Wang, Z. Qin, S. Zhang, and C. Zhang, "Cost-sensitive classification with inadequate labeled data," Inf. Syst., vol. 37, no. 5, pp. 508–516, 2012.
[9] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multilabel learning for image classification," IEEE Trans. Cybern., vol. 46, no. 2, pp. 450–461, Feb. 2015.
[10] X. Wu et al., "Top 10 algorithms in data mining," Knowl. Inf. Syst., vol. 14, no. 1, pp. 1–37, 2008.
[11] K. Zhang, X. Gao, D. Tao, and X. Li, "Single image super-resolution with multiscale similarity learning," IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 10, pp. 1648–1659, Oct. 2013.
[12] D. Landgrebe, "A survey of decision tree classifier methodology," IEEE Trans. Syst., Man Cybern., vol. 21, no. 3, pp. 660–674, May 1991.
[13] H. Liu, X. Li, and S. Zhang, "Learning instance correlation functions for multi-label classification," IEEE Trans. Cybern., vol. 47, no. 2, pp. 499–510, Feb. 2017.
[14] S. Zhang, "Parimputation: From imputation and null-imputation to partially imputation," IEEE Intell. Inform. Bull., vol. 9, no. 1, pp. 32–38, Jan. 2008.
[15] G. Góra and A. Wojna, "RIONA: A classifier combining rule induction and k-NN method with automated selection of optimal neighbourhood," in Proc. ECML, 2002, pp. 111–123.
[16] B. Li, Y. W. Chen, and Y. Q. Chen, "The nearest neighbor algorithm of local probability centers," IEEE Trans. Syst., Man, B, vol. 38, no. 1, pp. 141–154, Feb. 2008.
[17] X. Zhu, H.-I. Suk, and D. Shen, "Multi-modality canonical feature selection for Alzheimer's disease diagnosis," in Proc. MICCAI, 2014, pp. 162–169.
[18] J. Wang, P. Neskovic, and L. N. Cooper, "Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence," Pattern Recognit., vol. 39, no. 3, pp. 417–423, 2006.
[19] U. Lall and A. Sharma, "A nearest neighbor bootstrap for resampling hydrologic time series," Water Resour. Res., vol. 32, no. 3, pp. 679–693, 1996.
[20] D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong, "KNN algorithm with data-driven k value," in Proc. ADMA, 2014, pp. 499–512.
[21] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu, "Missing value estimation for mixed-attribute data sets," IEEE Trans. Knowl. Data Eng., vol. 23,no. 1, pp. 110–121, Jan. 2011.

[22] H. A. Fayed and A. F. Atiya, "A novel template reduction approach for the K-nearest neighbor method," IEEE Trans. Neural Netw., vol. 20,no. 5, pp. 890–896, May 2009.

[23] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao, "Linear cross-modal hashing for efficient multimedia search," in Proc. ACM MM, 2013, pp. 143–152.

[24] H. Wang, "Nearest neighbors by neighborhood counting," IEEE Trans.Pattern Anal. Mach. Intell., vol. 28, no. 6, pp. 942–953, Jun. 2006.

[25] Q. Liu and C. Liu, "A novel locally linear KNN method with applications to visual recognition," IEEE Trans. Neural Netw. Learn. Syst., to be published.

[26] X. Zhu, S. Zhang, J. Zhang, and C. Zhang, "Cost-sensitive imputing missing values with ordering," in Proc. AAAI, 2007, pp. 1922–1923.

[27] J. Hou, H. Gao, Q. Xia, and N. Qi, "Feature combination and the Knn framework in object classification," IEEE Trans. Neural Netw. Learn. Syst., vol. 27, no. 6, pp. 1368–1378, Jun. 2016.

[28] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for kNN classification," ACM Trans. Intell. Syst. Technol., vol. 8, no. 3, pp. 1–19,2017.