A Centroid *k*-Nearest Neighbor Method

Qingjiu Zhang and Shiliang Sun

Department of Computer Science and Technology, East China Normal University,
500 Dongchuan Road, Shanghai 200241, P.R. China
qjzh08@gmail.com, slsun@cs.ecnu.edu.cn

Abstract. k-nearest neighbor method (kNN) is a very useful and easy-implementing method for real applications. The query point is estimated by its k nearest neighbors. However, this kind of prediction simply uses the label information of its neighbors without considering their space distributions. This paper proposes a novel kNN method in which the centroids instead of the neighbors themselves are employed. The centroids can reflect not only the label information but also the distribution information of its neighbors. In order to evaluate the proposed method, Euclidean distance and Mahalanobis distance is used in our experiments. Moreover, traditional kNN is also implemented to provide a comparison with the proposed method. The empirical results suggest that the propose method is more robust and effective.

Keywords: Distance metric learning, k-nearest neighbor, Euclidean distance, Mahalanobis distance.

1 Introduction

Distance metric learning is to learn a distance metric from given examples. In recent years, distance metric learning is theoretically and empirically proved to be able to significantly improve the learning performance. Considerable research has been conducted on it. From the view of availability of the training outputs, it can be divided into two categories: unsupervised and supervised distance metric learning [1]. The supervised distance metric learning can be further divided into two categories, global and local metric learning. k nearest neighbor method (kNN) is one of the famous local metric learning approaches.

Though kNN classifies the query point according the k nearest neighbor examples, it also has its assumptions. Suppose $\{x_1, x_2, ..., x_k\}$ are the k nearest neighbors of the query point x. The label of x will be assigned to the most frequent class among those neighbors. This method is theoretically and empirically supported by early researchers' work [2], [3]. Lots of new methods focused on kNN have been proposed in recent years [4], [5], [6], [7]. There are also some other algorithms that employ the labels of k nearest neighbors to weight classifiers participating in the classification [8]. However, kNN has the assumption that the class conditional probabilities of the local nearest neighbors is a constant [9], [10]. This assumption can be regarded as that the class conditional

L. Cao, J. Zhong, and Y. Feng (Eds.): ADMA 2010, Part I, LNCS 6440, pp. 278–285, 2010.

probabilities are distributed smoothly among neighbors. Moreover, different setting of the parameter k may lead to quite different results, which makes the kNN quite unstable for k.

It is clear that all the information used to identify the query point x just comes from the k nearest neighbors. We can assume that around x there exists a region in which only k neighbors are included. However, there are a variety of distributions of those k neighbors. They may be around the query point, or at a side of the query point. The traditional kNN does not reflect this information at this point. Moreover, the hypothesis is often expected by a majority voting measurement which is not reliable due to few neighbors. Consequently, the performance will be improved if the above problems are solved.

In this paper, a novel kNN method based on centroids is proposed. The distribution of k nearest neighbors is properly taken into consideration. Real training examples are not directly used to identify the query points. Instead, centroids generated by real examples with probabilistic labels are used to predict. Thus, we further suggest our new method: centroid k-nearest neighbor method. In order to evaluate the proposed method, traditional kNN is also implemented together with the proposed method on 12 classification problems.

The rest of this paper is organized as follows. In Section 2, how to create the centroids is stated in detail. The novel method and its algorithm are presented in Section 3. In Section 4, experimental results involving 12 real-world problems are reported. At last, conclusions are stated in Section 5.

2 Centroids for kNN

Centroids can reflect the distribution and label information of the neighbors, which can be used to further improve the performance of kNN.

2.1 C-Means Clustering

In classification problems, lots of methods devote themselves to enlarging the distance between classes while narrowing the distance within classes. There are lost of this kinds of algorithms used in both supervised and unsupervised learning. For instance, the Fisher linear discriminant analysis method [11] project the original space into a new space in which distance between classes is enlarged while distance within class is narrowed. Some clustering methods also cluster unlabeled examples according to the distances between class and within class [12], [13].

The c-means clustering is an unsupervised method which assigns each point to the cluster whose center (also called centroid) is the nearest [14]. The center is the average of all the points in the cluster. That is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster. The center is called centroid or center example in this paper. Obviously, centroids play an important role in the c-means clustering. And it can briefly reflect the local distribution information. It will help if this kind of information is taken into consideration to learn supervised problems.

2.2 Classification with Centroids

Centroids are combined with kNN to solve classification problems in this paper. If the centroid of a query point is closer to that query point, it will be more creditable to predict the query point. Therefore, in the proposed method the query example is predicted from the centroids of its neighbors instead of directly from its neighbors. The centroid is constructed by the nearest neighbors according to

$$x'_k = \frac{1}{k} \sum_{j=1}^{k} x_j, \tag{1}$$

where x'_k denotes the centroid of the k nearest neighbors of the query point x. Thus, j nearest neighbors will construct a j-th centroid. Consequently, k centroids will be constructed. Fig. 1 shows an instance of different centroids constructed by different numbers of neighbors. The figure illustrates that a smaller number of nearest neighbors may not play as good as a larger number of nearest neighbors.

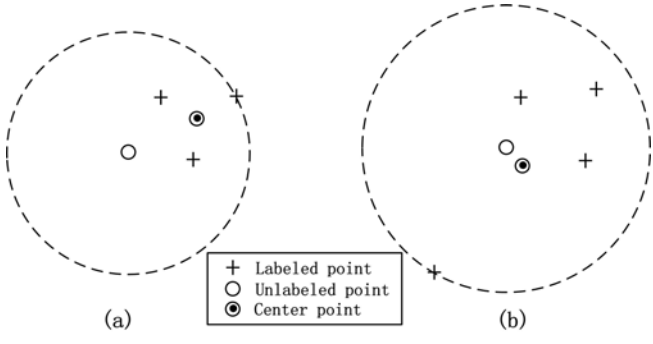
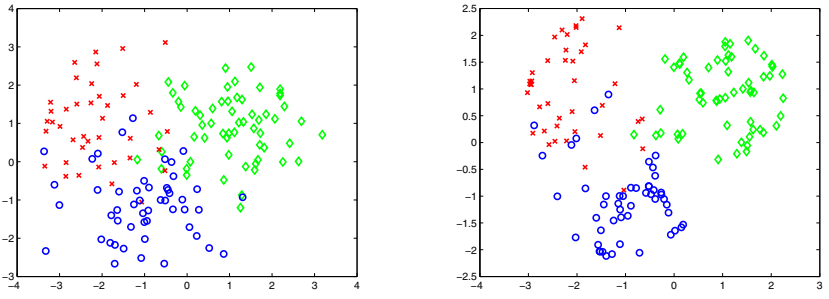


Fig. 1. (a) and (b) show the centroids of the three nearest neighbors and four nearest neighbors

Fig. 2 shows the distributions of a two dimensional toy data and its representation of the centroids. The original dataset has three classes which are drawn from a Gaussion distribution respectively. If each point is substitute by the center of nine nearest neighbor, the data distribution will appear quite differently. In the figure we can find that examples belonging to the same class are clustered together. In other words, distances between classes are enlarged while distances within class are narrowed.

If the centroids are used, two problems may arise.

- The neighbors corresponding to the nearest centroid may not belong to the same class.
- The centroids may become closer to the query point as the parameter k becoming very large.



(a) (b)

Fig. 2. (a) and (b) respectively show the distribution of the original data and the distribution of the centroids constructed by the nine nearest neighbors

To overcome this two problems, we defined a new distance d_c in which not only the distance information but also the label information are involved. The form of d_c is

$$d_c = c_1 + \alpha c_2, \tag{2}$$

where c_1 and c_2 are the distance and labeled information respectively. α is a balance parameter. c_1 and c_2 are calculated according to

$$c_1 = e^{-\frac{d^2}{\sigma^2}}, \quad \sigma = ||(d'_1, ..., d'_k)^T||_2 \tag{3}$$

and

$$c_2 = \sum_{i=1}^{t} p_i \log_t(p_i), \quad p_i = \frac{w_i + \frac{1}{kt}}{\sum_{j=1}^{t} w_j + \frac{1}{k}} = \frac{ktw_i + 1}{t(k^2+1)}, \tag{4}$$

where d'_k means the real distance between the query point and the center of its k nearest neighbors. And this kind of distance measurement can be Euclidean distance and Mahalanobis distance, etc. t and w_i denote the number of classes and the number of neighbors belonging to class j respectively. c_1 is a kind of Gaussian function to calculate the confidence of the nearest neighbors, which ensures the closer neighbor has a larger value. It assumes the query point is the center of the around neighbors. Therefore, the parameter σ should be the norm of the distances between the neighbors and the query point. c_2 utilizes entropy to ensure that the centroid with more confident label have a larger value and a priority to be selected. Thus, the centroid with the largest d_c will be the most creditable one.

3 Centroid *k*NN

Based on the strategy of centroids, a centroid k-nearest neighbors method (CkNN) can be deduced. When centroids are used in kNN, k centroids will be

constructed according the strategy of the above section. The label of the query point will be assigned to the label of its nearest centroid. However, its label can also be decided by the k nearest centroids by a majority voting method. That means those constructed points are implemented under a kNN again to predict the final hypotheses. Noting that the k of the second kNN is no larger than the value of k of the first kNN. Now the algorithm of the CkNN can be described in pseudocode in Table 1.

Table 1. The algorithm of CkNN

Given:
Labeled training set L and unlabeled test set U,
The parameters k and k' of the first and second kNN,
For each example x **in** U**:**
Calculate the Euclidean distance from each training example to x,
Find out the k nearest neighbors X=$\{x_1, x_2, ...x_k\}$,
Calculate the k centroids corresponding to each group nearest neighbors,
Calculate the components c_1 and c_2 for each centroid,
Calculate the distance d_c from x to each centroid,
Identify the x using k' nearest neighbors method.

4 Experiments

Datasets used in the experiments are from UCI [1] which is public and commonly used as benchmarks by scientists in the field of machine learning. Twelve datases involving different domains are used in this paper. The parameter k for the kNN ranges from one to nine.

In order to evaluate the proposed method, two kinds of distance measurements are launched, and they are Euclidean distance and Mahalanobis distance. For a n–dimensional problem, the Euclidean distance between two examples x and m_i is calculated according to

$$d(x, m_i) = ||x - m_i|| = \sqrt{\sum_{j=1}^{n} (x(j) - m_i(j))^2}, \tag{5}$$

where $x(j)$ denotes the j-th feature value. However, the form of Mahalanobis distance is

$$d(x, m_i) = \sqrt{(x - m_i)^T \sum^{-1} (x - m_i)}, \tag{6}$$

where $\sum$ denotes the covariance matrix of the training set.

[1] http://archive.ics.uci.edu/ml/

The set of experiments is carefully designed. The comparison between CkNN and traditional kNN is implemented on all used datasets. Moreover, in order to clearly reflect the effectiveness of the proposed method, all the experiments are implemented under a ten fold cross-validation (CV) method. In the proposed method, there is a parameter α. Its value is also calculated under a CV method on the training set.

4.1 Euclidean Distance

Table 2 shows the classification results (in accuracy) of the experiments when Euclidean distance is calculated. It is clear that the proposed method has a smaller variance on the k results, which reflects the proposed method is more stable and robust. The CkNN also has a higher accuracy on the mean value. Moreover, when the value of k is set to one (Nearest neighbor method), we can find that the proposed method is better than the traditional method almost on all the datasets.

Table 2. The empirical results when Euclidean distance is used

D	Method	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	Mean	Var	Max
1	kNN	73.6	65.2	68.8	67.6	66.4	68.8	70.4	70.0	68.4	68.8	0.024	73.6
1	CkNN	73.6	67.2	68.4	68.0	70.0	68.0	68.4	67.6	69.6	68.9	0.019	73.6
2	kNN	71.2	67.2	67.2	66.4	65.2	64.8	61.6	62.4	60.0	65.1	0.034	71.2
2	CkNN	72.8	70.0	69.2	68.8	67.6	66.4	66.8	65.6	65.6	68.0	0.023	72.8
3	kNN	87.5	81.4	85.6	82.2	84.4	81.7	83.3	82.2	83.3	83.5	0.020	87.5
3	CkNN	88.1	82.5	84.2	83.3	84.4	82.8	83.6	82.8	83.1	83.9	0.017	88.1
4	kNN	96.0	94.7	96.0	96.0	95.3	94.7	95.3	96.0	96.7	95.6	0.007	96.7
4	CkNN	95.3	95.3	96.7	96.0	95.3	95.3	95.3	94.7	94.7	95.4	0.006	96.7
5	kNN	83.6	84.6	84.2	84.2	84.2	84.2	82.1	81.7	80.8	83.3	0.014	84.6
5	CkNN	87.9	85.4	85.0	84.2	84.6	83.8	84.6	83.3	82.5	84.6	0.015	87.9
6	kNN	60.0	50.0	68.3	68.3	61.7	60.0	58.3	51.7	46.7	58.3	0.076	68.3
6	CkNN	63.3	53.3	70.0	63.3	63.3	66.7	63.3	61.7	60.0	62.8	0.046	70.0
7	kNN	80.8	81.8	84.4	84.9	85.6	86.4	86.9	85.9	86.4	84.8	0.022	86.9
7	CkNN	82.3	82.8	85.4	83.9	85.4	84.9	87.2	86.9	86.7	85.0	0.018	87.2
8	kNN	60.0	63.3	66.7	67.4	67.4	66.7	67.4	66.3	66.7	65.8	0.025	67.4
8	CkNN	65.2	67.0	66.7	68.5	69.3	69.3	68.2	67.0	66.7	67.5	0.014	69.3
9	kNN	55.0	44.4	41.3	40.6	39.4	42.5	38.1	36.3	33.8	41.3	0.061	55.0
9	CkNN	56.9	43.1	50.0	47.5	43.8	42.5	42.5	41.3	41.9	45.5	0.051	56.9
10	kNN	90.3	92.0	91.5	91.9	93.1	92.6	92.9	92.5	93.4	92.2	0.009	93.4
10	CkNN	92.5	91.7	92.3	92.2	92.5	92.3	92.6	92.5	92.8	92.4	0.003	92.8
11	kNN	98.2	95.5	94.6	89.1	86.4	85.5	82.7	80.9	80.0	88.1	0.067	98.2
11	CkNN	98.2	95.5	97.3	96.4	93.6	89.1	89.1	89.1	86.4	92.7	0.044	98.2
12	kNN	82.7	81.2	84.0	84.0	85.2	86.7	88.5	87.8	88.8	85.4	0.027	88.8
12	CkNN	86.7	85.8	87.8	88.1	87.9	87.9	87.8	87.6	87.6	87.5	0.007	88.1

4.2 Mahalanobis Distance

Table 3 presents the results of the proposed under the Mahalanobis distance. In the 12 classification problems, we can find that CkNN has a smaller variance

value over the ks, which means that CkNN palys more stably than the traditional kNN. As to the mean value, CkNN outperforms kNN on almost all the datasets. Moreover, when k=1 (Nearest neighbor method) CkNN still runs better than kNN. Although CkNN is not as good as kNN on few datasets, but it outperform kNN in the majority of all the datasets. The above results are sufficient to show that the proposed method is better than kNN.

Table 3. The empirical results when Mahalanobis distance is used

D	Method	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	Mean	Var	Max
1	kNN	93.6	86.0	93.6	92.8	94.0	91.6	93.2	92.8	93.6	92.4	0.025	94.0
1	CkNN	93.2	88.0	95.6	95.6	96.4	94.4	94.4	94.0	94.4	94.0	0.025	96.4
2	kNN	67.2	64.4	69.2	67.2	64.8	64.4	62.4	62.8	62.4	65.0	0.024	69.2
2	CkNN	70.8	66.4	70.0	70.4	70.0	67.2	66.4	66.8	66.0	68.2	0.020	70.8
3	kNN	63.9	63.9	63.9	63.9	63.9	63.9	63.9	63.9	63.9	63.9	0.000	63.9
3	CkNN	63.9	63.9	63.9	63.9	639	63.9	63.9	63.9	63.9	63.9	0.000	63.9
4	kNN	92.0	90.7	91.3	90.0	90.7	88.7	88.0	86.7	87.3	89.5	0.019	92.0
4	CkNN	92.0	92.0	93.3	92.0	91.3	90.0	89.3	88.7	88.0	90.7	0.018	93.3
5	kNN	90.4	92.1	89.6	86.7	87.1	88.3	87.9	84.2	85.4	88.0	0.025	92.1
5	CkNN	93.3	92.1	91.3	90.8	90.0	89.2	88.8	88.3	88.3	90.2	0.018	93.3
6	kNN	83.3	60.0	68.3	66.7	63.3	63.3	61.7	60.0	53.3	64.4	0.083	83.3
6	CkNN	80.0	60.0	75.0	73.3	70.0	70.0	70.0	70.0	66.7	70.6	0.055	80.0
7	kNN	75.1	76.2	79.5	80.0	82.1	79.5	80.8	80.0	81.8	79.4	0.024	82.1
7	CkNN	78.7	77.7	79.2	79.2	80.8	81.5	81.8	81.5	82.1	80.3	0.016	82.1
8	kNN	74.1	72.6	79.6	78.5	80.7	79.6	82.2	78.9	79.6	78.4	0.031	82.2
8	CkNN	75.2	74.1	77.4	75.9	80.0	80.0	79.6	80.7	80.4	78.2	0.025	80.7
9	kNN	61.3	52.5	45.6	50.0	51.3	50.0	51.9	52.5	53.1	52.0	0.041	61.3
9	CkNN	63.8	53.1	51.9	52.5	51.9	53.8	53.1	51.3	51.9	53.7	0.039	63.8
10	kNN	83.4	78.0	82.0	78.6	81.5	76.9	80.6	76.2	79.1	79.6	0.024	83.4
10	CkNN	83.7	78.0	79.4	78.8	80.5	79.5	80.2	79.4	79.5	79.9	0.016	83.7
11	kNN	94.6	90.0	91.8	87.3	84.6	80.0	76.4	75.5	75.5	83.9	0.074	94.6
11	CkNN	94.6	88.2	90.9	90.0	90.0	86.4	87.3	87.3	87.3	89.1	0.026	94.6
12	kNN	78.2	77.8	83.3	82.7	86.4	87.0	88.2	89.1	89.4	84.7	0.045	89.4
12	CkNN	85.1	82.4	86.9	86.0	87.6	87.3	88.2	88.2	88.1	86.6	0.019	88.2

5 Conclusions

This paper proposed a new kNN. The real data are not directly used to calculate the distance. Instead, they are substituted by the centroids of the nearest neighbors. The empirical results illustrates that the proposed method can significantly improve the effectiveness of kNN.

Further investigations on centroids are possible. In this paper, centroids are used to replace real neighbors to solve the classification problems. From the results we can conclude that centroids could afford as sufficient information as the real points for the problems. Consequently, all the examples (including training dataset and test dataset) replaced by their centroids may be regarded as an additional view of the original dataset. If the additional view could provide

complementary information with the original view, centroids will become a view-creating method for the multiple-view learning problems.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Projects 60703005 and 61075005.

References

1. Yang, L.: Distance Metric Learning: A Comprehensive Survey. Department of Computer Science and Engineering. Michigan State University (2006)
2. Cover, T., Hart, P.: Nearest Neighbor Pattern Classification. IEEE Transaction on Information Theory, 21–27 (1967)
3. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley, New York (1973)
4. Achtert, E., Kriegel, H.P., Kröger, P., Renz, M., Züfle, A.: Reverse k-Nearest Neighbor Search in Dynamic and Deneral Metric Databases. In: Proceedings of the 12th International Conference on Extending Database Technology, pp. 886–897 (2009)
5. Sun, S., Huang, R.: An Adaptive k-nearest Neighbor Algorithm. In: Proceedings of the 7th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 91–94 (2010)
6. Weinberger, K.Q., Saul, L.K.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. Journal of Machine Learning Research 10, 207–244 (2009)
7. Jin, C., Guo, W.: Efficiently Monitoring Nearest Neighbors to a Moving Object. In: Proceedings of International Conference on Advanced Data Mining and Applications, pp. 239–251 (2007)
8. Sun, S.: Local Within-class Accuracies for Weighting Individual Outputs in Multiple Classifier Systems. Pattern Recognition Letters 31(2), 119–124 (2010)
9. Hastie, T., Tibshirani, R.: Discriminant Adaptive Nearest Neighbor Classification. IEEE Pattern Analysis and Machine Intelligence 18(6) (1996)
10. Friedman, J.: Flexible Metric Nearest Neighbor Classification. Stanford University Statistics Department, Tech. Rep. (1994)
11. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley and Sons, New York (2001)
12. Ng, A.Y., Jordan, M.I., Weiss, Y.: On Spectral Clustering: Analysis and An Algorithm. In: Advances in Neural Information Processing Systems (2001)
13. MacKay, D.J.C.: Information Theory, Inference and Learning Algorithms. Cambridge University Press, Cambridge (2003)
14. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An Efficient K-means Clustering Algorithm: Analysis and implementation. IEEE Transaction on Pattern Analysis and Machine Intelligence 24, 881–892 (2002)