Farhad M. Kazemi **Machine Learning**

Title:Coarse to fine K nearest neighbor classifier [1]

a) Coarse to Fine K Nearest Neighbor Classifier Algorithm

1- **At first, we are faced with determining n training instances that are close to the test instance coarsely.**
2- **In the next phase, we select K(K<=n) nearest neighbors of the test instance from the n training instances in step 1.**
3- **Then, we classify the test instance using the class labels of the K nearest neighbors.**
4- **End**

Three steps explained step by step as following.

1-This algorithm actually wants to present test instance Y as a linear combination of all the training instances as following.

$$Y = \sum_{i=1}^{N} \gamma_i X_i \qquad (1)$$

$$Y = X\gamma \qquad (2)$$

Where $\gamma = (\gamma_1 \dots \gamma_N)^T, X = (X_1 \dots X_N)$

Now, we have to solve Equation (2). For solving this equation, we use the **Lagrangian algorithm**. For more details please refer to **section ****.

So based on the Lagrangian algorithm, $\boldsymbol{\hat{\gamma} = (X^T X + uI)^{-1} X^T Y}$ (I is the identity matrix)

So, our classifier can calculate $e_i = \|Y - \hat{\gamma_i} X_i\|^2$ where $\hat{\gamma_i}$ is i-th entry of $\hat{\gamma}$.

Now, our classifier selects n training instances that have the first n smallest $\boldsymbol{e_i}$ and mark them as $\boldsymbol{Z_1 \dots \dots Z_n}$, in sequence.

2- Now, our classifier, for expressing test instance Y, exploits a weighted sum of $\boldsymbol{Z_1 \dots \dots Z_n}$**.**

$$Y = \sum_{i=1}^{n} w_i Z_i \qquad (3)$$

$$Y = Zw$$

$$w = (w_1 \dots w_n)^T, Z = (Z_1 \dots Z_N)$$

So for solving this equation (based on Lagrangian algorithm), we have

$\boldsymbol{\hat{w} = (Z^T Z + uI)^{-1} Z^T Y}$

(μ is positive constant)

(I is the identity matrix)

Now, CFKNN classifier calculates $d_i = \|Y - \widehat{w_i} Z_i\|^2$ as the similarity metric between $Z_i$and Y. Where $\widehat{w_i}$ is i-th entry of $\hat{w}$.

It is clear that a smaller d_i considers a higher similarity between Y andZ_i.

In this step, our classifier chooses K training instances that have the first K smallest $\boldsymbol{d_i}$ from $\boldsymbol{Z_1 \dots \dots Z_n}$ and marks them in order as $\boldsymbol{S_1 \dots \dots S_K}$.

3-Counting the number of the training instances from the j-th class. (j=1,…,c)

c is the number of classes

m_j is the number of training instances from the j-th class

As we know, $K = \sum_{j=1}^{c} m_j$

In this step, we have to find m_j that is maximum (t- th class) and then our classifier assigns the test instance Y into the t-th class.

4-End

======================

Note: In the procedure of CFKNN classifier, excluding the training instances that are far from the test instance and reaching the best(optimal) nearest neighbors of the test instance are the most important causes for obtaining a higher accuracy in comparison to KNN classifier.

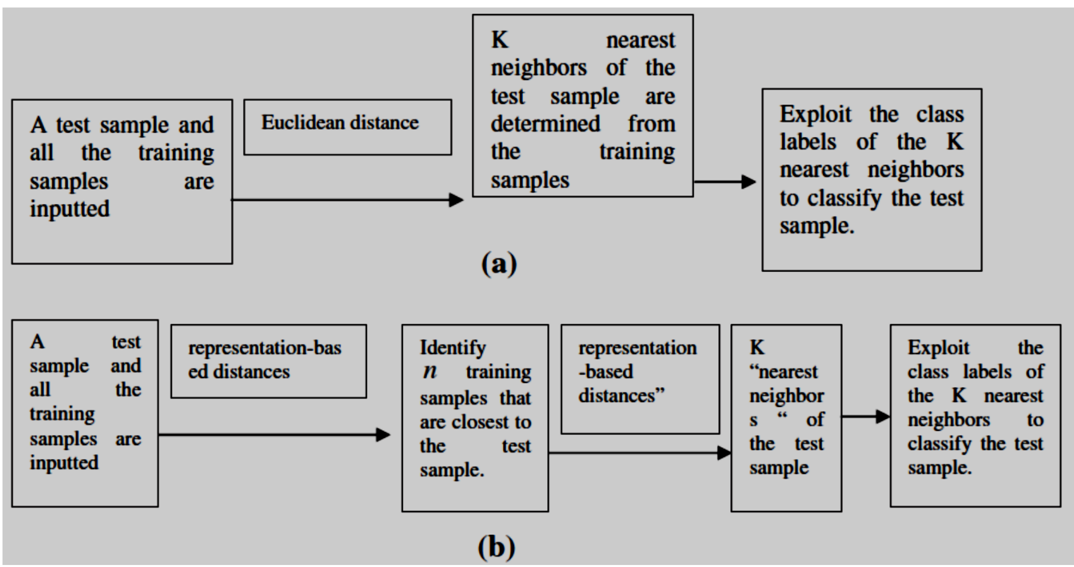

Figure 1. (a) Flowchart of KNN and (b) Flowchart of CFKNN.

Section **: Lagrangian algorithm for solving Equation (2)

We know that $X\gamma$ best approximate Y, so we can minimize the equation of $||Y - X\gamma||^2$… by regulating γ.

According to the theories of numerical analysis, we know that if **n** has a small norm, the solution of Equation (2) can generalize well. So, it would be expected that $||\gamma||^2$ be as low as possible.

So, we can define **Lagrangian function** as following.

$f(\gamma) = ||Y - X\gamma||^2 + \mu||\gamma||^2$ (μ is positive constant)

Based on the information regarding the theories of numerical analysis, minimizing $f(\gamma)$ and satisfying $\frac{\partial f(\gamma)}{\partial \gamma} = 0$ can help us to obtain the optimal solution. On the other hand, we have

$$\frac{\partial f(\gamma)}{\partial \gamma} = 0$$

$$2(X^T X + \mu I)\gamma = 2X^T Y$$

So we have $\boldsymbol{\hat{\gamma} = (X^T X + uI)^{-1} X^T Y}$ (I is the identity matrix)

b)

The flowcharts below show some principal ideas regarding K Nearest Neighbor classifier (KNN) and Coarse to fine K nearest neighbor classifier (CFKNN). With a brief look, we can clearly see that the main similarity between KNN and CFKNN is that both of them can classify a test instance by using the labels of the K nearest neighbors of the test instance.

Regarding the main differences between KNN classifier and CFKNN classifier, I should say that one of the most important differences is that KNN classifier exploits the Euclidean distance for determining the nearest neighbors of the test instance and CFKNN classifier uses representation distances to do so. It is clear that CFKNN has more steps in comparison to KNN.

Also, CFKNN classifier considers that a number of training instances that can provide a good approximation for the test instance must be chosen as its nearest neighbors. So this method involves representative training instances. Actually mentioned method tries to select the training instances that have much portion to this approximation as the nearest neighbors of the test instance. So all of the training instances compete to each other for winning. It is clear that these K training instances can make a good approximation for the test instance. So, we can conclude that this approximation includes useful information without waste information. Based on the given information, a good approximation of the test instance can conclude to high classification accuracy. On the other hand, KNN classifier calculates the similarity between the test instance, and each training instance that maybe the nearest neighbors includes of much more unneeded information to exhibit the test instance.

For evaluating CFKNN classifier in comparison to the KNN classifier, Yong Xu and et al. [1] obtained a smaller mean of the cosine correlation coefficients from different nearest neighbors for their algorithm. This result shows that the nearest neighbors obtained using CFKNN contain less waste information in comparison to those obtained by KNN classifier. So, the weighted sum of the nearest neighbors obtained by CFKNN classifier can better approximate the test instance in comparison to KNN classifier.

c)

To validate CFKNN method on the classification performance, the experiments are conducted on Palmprint recognition and face recognition applications. The increasing demand for personal identification is calling for more convenient and secure systems than traditional methods, i.e. passwords, ID cards, which could be forgotten or lost occasionally. Biometrics, identification/verification of a person by the physiological or behavioral characteristic, is playing an important role in modern personal identification systems. More and more biometric features are proposed and used in commercial systems, such as fingerprint, palmprint, facial feature, iris, etc. However, there is not a perfect biometric method that can suffice all the situations. For example, fingerprint is the most widely used biometric feature, but some workers have so bad fingerprints that can't be recognized well. I think the main reason that Yong Xu and et al. tested their algorithm on palmprint recognition is that, in comparison to another biometric authentication methods, palmprint recognition is one of the most user-friendly and reliable methods. Palmprint is concerned with the inner surface of the hand. It is unique between people, even palms of one single person's two hands or twins' palmprints.

d)

Palmprint dataset :

The Palmprint system, which is the multispectral palmprint recognition system, uses multispectral capture device to sense images under different illuminations, including Red, Green, Blue and Infrared.

This paper uses Hong Kong Polytechnic University palmprint dataset. As we know, Green, Red and Blue could composite different light in visible spectrum. Moreover, most of color images are recorded or represented by these three colors. Infrared light could offer pictures with more penetrability. Actually, this paper built a multispectral system using these 4 illuminations.

Palmprint images were collected from 250 persons (55 women and 195 men). Since the two palmprints (right-hand and left-hand) of each person are different, it was captured both and treated them as palmprints from different people. So there are 500 palms. Moreover, this process accomplished in two sessions. Actually, in each session, every palm provided 6 palmprint images at each illumination. So, there are 6000 palmprint images for each of Red, Green, Blue and Infrared illuminations. Then the dataset has 24,000 palmprint images in total. Also, the resolution of the images is 352 *288.

In this paper the 128*128 region of interest (ROI) domain was extracted from each palmprint image. For ROI images of a palm captured under each illumination, first of all the first three images captured in the first session were exploited as training instances (150 instances) and all the images captured in the second session were exploited as test instances. Then, the first six images took in the first session were used as training instances (300 instances) and all the images captured in the second session were used as test instances. In this paper, each ROI image resized to a 32*32 image and converted into a one-dimensional unit vector with length of 1.

Eventually, we can clearly see that this algorithm will reach a higher accuracy when we have more training instances. Moreover, it is clear that when the number of training instances varies from 75 to 300, the accuracy of CFKNN algorithm changes a little.

e)

Based on the given information in this paper, I saw that CFKNN algorithm always obtains a higher classification accuracy than KNN. As you can see in Figure 2, when only the one nearest neighbor was used for classification, the classification accuracies of CFKNN on the blue, green, red and near infrared illumination images are 98.37%, 98.17%, 97.67% and 95.27%, in sequence. In addition, as shown in Figure 2, the classification accuracies of KNN on the blue, green, red and near infrared illumination palmprint images are 95.17%, 92.07%, 95.03% and 94.77%, in order. As you see, CFKNN algorithm also outperforms NFL (nearest feature line), NFS (nearest feature space), CBNNC (center-based nearest neighbor classifier) and NNLC (Nearest neighbor line classifier) algorithms. For more details, please refer to [1]. Also, Both CFKNN and KNN obtain the best performance with K = 1(K is the number of nearest neighbor). The main reason is that palmprint images usually have no much deformation and the ‘‘nearest neighbor’’ of the test instance has a very high probability of being from the same palm as the test instance.

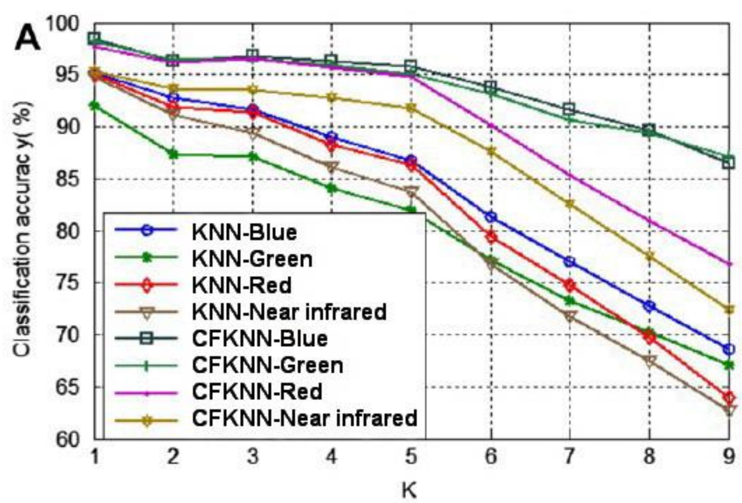

Figure 2. Results of CFKNN and KNN classifiers on the multispectral palmprint image dataset. The first three images captured in the first session were exploited as training instances (150 instances) and all the images captured in the second session were exploited as test instances.

Reference

Xu, Yong, et al. "Coarse to fine K nearest neighbor classifier." *Pattern Recognition Letters* 9.34 (2013): 980-986.