Classification Accuracy and Model Selection in *k*-Nearest Neighbors Classifiers for Data Driven Learning

Jing Lu

Xinxiang University, Henan, China
E-Mail:Jinger@163.com

***Abstract* - Pattern classification is a core research area and a main task in pattern recognition. A classifier induced by machine learning algorithms maps an unlabeled instance to a label using internal data structures. In this paper we experiment first by changing the *k* value of nearest neighbors from 3 to 15 and compare the accuracy of two classifiers on various training and test sets. The results show that the attribute-weighted *k*NN classifier is better than others when the dataset has low dimensionality. The experimental results show that cross validation is good for improving classification accuracy with or without 3-fold cross validation for adjusting the weights. This is due to the cross validation accuracy estimator depending on the two factors of the training set and the partition into fold. Then we select nearest neighbors and change fold number from 3 to 10, the size of every fold decreases when the number of folds increase, the accuracy of classifier change, but instance-weighted *k*NN is relatively stable. Thus, we can see the sensitivity of *k*-cross validation in accuracy estimation.**

Keywords - pattern classification, accuracy estimation, stability, k-fold cross validation, k-nearest neighbor.

I. INTRODUCTION

For selecting a suitable classifier from a set of classifiers on a given dataset, we often rely on the classification accuracy of the classifiers. Firstly, We review the formalism of pattern classification problem, various *k* nearest neighbor classifiers and the methods of accuracy estimation. For attribute-weighted *k*NN classifier and instance-weighted *k*NN classifier, we report on a large-scale experiment to estimate the effects of different parameters on these algorithms on real-world datasets. Generally, the classification methods are divided into supervised and unsupervised. The *k*-nearest neighbors classifier (*k*-NN) is a classical supervised method based on statistical data, which maps any feature vector *X* to the pattern class that appears most frequently among the *k*-nearest neighbors.

The performance of a classifier depends on the interrelationship between sample size, number of features, and classifier complexity. Estimating the accuracy of a classifier is important not only to predict its future prediction accuracy, but also for choosing a good classifier from a given set. In training phases of those supervised classifiers, accuracy estimation is helpful for adjusting partial parameters until achieving expected accuracy rate. The accuracy of a classifier is the probability of correctly classifying a random or selected instance. Usually, the absolute accuracy is unknown, cannot be calculated, and must be estimated from given datasets.

Various methods have been applied to estimate the accuracy rate, such as holdout, leave-one-out, *k*-fold cross-validation (*k*-cv) and bootstrap. The two most common methods probably is *k*-cv and bootstrap. Kohavi has demonstrated that ten-fold stratified cross- validation is the best method for model selection through comparing cross-validation and bootstrap, even if computation power allows using more folds. The experimental details refer to [1].

It’s well known that no accuracy estimation can be correct all the time. In this paper, we just are interested in analyzing the contributions of *k*-fold cross-validation for adjusting weights in various weighted *k*-NN algorithms. We don’t discuss problems such as feature selection, prediction and sample pruning, which also are one of the most important and difficult problems in improving the accuracy of *k*-nearest neighbors classifier. Our approach is primarily inspired by two sources. One is the work of Juan Diego et al [2], who analyzed the sensitivity to changes in the training set and the sensitivity to changes in the folds. The other is the work of Naonori and Ryohei [3], who give a comparison of cross-validation and bootstrap for estimating expected error rates of neural network classifiers.

The rest of the paper is organized as follows: In Section 2, we briefly explain the used methods to improve and to estimate the accuracy of classifiers. And lists main methods for accuracy estimation. In Section 3, we review the related works about improving classification accuracy and model selection. In Section 4, we elaborate the experimental process, such as k-fold cross validation for adjusting the weight for *k*NN. In Section 5, we discuss both the experiment results and others previous works. Finally, We conclude a summary in Section 6.

II. METHODOLOGY

The pattern classification problem is generally formulated as follows: an input pattern is to be assigned to one of classes based on a vector of features, the feature vector. The recognition system is operated in two modes: training (or learning) and classification (or testing). In the testing mode, the trained classifier assigns the input pattern to one of the pattern classes under consideration based on the measured features. In statistical pattern recognition, the features are assumed to have a probability density or mass function conditioned on the pattern class. Thus, a pattern vector belonging to class is viewed as an observation drawn randomly from the class-conditional probability function.

A. Weighted k-Nearest Neighbor Classifiers

The nearest neighbor classifier is the simplest pattern classification algorithm yet devised. In *k*-nearest neighbors classifiers, each input pattern is mapped into class according to the majority of the labels of its *k*-nearest neighbors in the training samples. Let be a set of training patterns whose classes are knows. Let be the sequence of patterns in which are closest to. The *k*-nearest neighbors of are defined as the labeled subset of feature vectors. If k is even, it is necessary to define an auxiliary procedure to handle ties in Equation (2). Generally, Ties are resolved by randomly selecting one of the tied pattern class, or considering fewer or more than nearest neighbors. In our experiment, we randomly select one of the tied pattern class to resolve the ties.

For improving the classification accuracy, some fundamental improvements are applied to *k*-NN classifier, such as weighted nearest neighbor classifiers (WKNN), where a weight for each training pattern is assigned and is used in the classification. The goal of weight functions is to cause distant neighbors to have less effect on the majority vote than the closer neighbors [4]. After selecting the Euclidean distance metric as the best one, two different weight functions are taken into consideration in our experiments. One is the attribute-weighted function of which every attribute is associated with a weight. We define the distance between two samples and as follow: or has attributes. The attribute-weighted -NN rule initializes every attribute weights with random values, and computes the distances to every sample in the training set, then selects the class which belongs to the majority of the nearest neighbors. The other is the instance-weighted function of which a weight is randomly generated for every instance (or sample). After calculating nearest neighbors of a sample according to Euclidean distance metric, the weights assigned to nearest neighbors are summed when the neighbors belongs to the same class,

Finally, the class value is assigned to the test pattern when the weight sum is the greatest value.

In training pattern, the *k*-fold cross-validation was performed for adjusting weights value until reaching desired classification accuracy. Then the weights are used in the classification pattern.

B. Accuracy Estimation of Classifiers

Estimating the accuracy of a classifier is important not only to predict its future prediction accuracy on test set, but also for choosing a classifier from a given set, even combining classifiers. The classification error or the error rate, is the ultimate measure of the performance of a classifier. For a given decision rule, if an analytical expression for the error rate was available, it could be used to study the behavior of as a function of the number of features, true parameter values of the densities, number of training samples, and prior class probabilities. But it is very difficult to obtain a closed-form expression for.

Assign input pattern to class for which the conditional risk is minimum, where is the loss incurred in deciding when the true class is , and is the posterior probability. However, the class-conditional densities are usually not known in practice. In this case, we must either estimate the density function or directly construct the decision boundary based on the training data, such as -nearest neighbors decision rules.

As a result, the performance of a classifier depends on both the number of available training samples as well as the specific values of the samples. Bayes error indicates the least possible value of error, which is achieved by the Bayes optimal classifier. Hence, the value of Bayes error forms a lower bound for the true error of any classifier designed for a set of training sample data generate for a specific distribution.

In practice, the accuracy rate of a recognition system must be estimated from all the available samples which are split into training and test set [5].Therefore, the generalization ability of a classifier refers to its performance in classifying test patterns which were not used during the training stage. The percentage of classified test samples is taken as an estimation of the accuracy rate. In order for this accuracy estimation to be reliable in predicting future classification performance, not only should the training set and the test set be sufficiently large, but the training samples and the test samples must be independent. In our experiments, we define the accuracy of the *k*-NN classifiers is the percentage of correctly classifying a randomly selected instance from a given data set.

C. Methods for Accuracy Estimation

There is always no better solution to divide the available samples into training and test sets. Therefore, different random splits will result in different error estimates. Various methods applied to estimate the accuracy rate are mainly different in how they utilize the available samples as training and test sets [4, 6].

Holdout method, sometimes called test sample estimation, partitions the data into two mutually exclusive subsets called a training set and a test set. So different partitioning will give different estimations. It is common to designate of the data as the training set and the remaining as the test set.

In leave-one-out method, a classifier is designed using () samples, and is evaluated on the one remaining sample. This is repeated times with different training set of size (). Estimate is unbiased but it has a large computational requirement.

Cross-validation method, sometimes called rotation estimation, is a compromise between holdout and leave-one-out methods. In -fold cross validation, the data set is randomly split into mutually exclusive subsets (or the folds) of approximately equal size. A classifier is learned using folds, and test on the remaining fold. The classifier is trained and tested times. Estimate has lower bias than the holdout method and is cheaper to implement than leave-one-out method.

Bootstrap method was introduced by Efron and is fully described in [7]. Given a dataset of size, many bootstrap samples are created by sampling from the data (with replacement). The accuracy estimate is derived by using the bootstrap samples. Bootstrap estimates can have lower variance than the leave-one-out method, more computation demanded, and being suit for small sample size situations. In improved classifiers, bootstrap method is shown to work well with nearest neighbor classifiers, researchers generate an artificial training set by applying the bootstrap method, meanwhile the bootstrapped training set instead of the original training set.

D. Fold Cross Validation Method

In *k*-fold cross-validation, a data set is at random partitioned into folds of similar size. Generally, researchers will suppose that is multiple of. Let be the complement data set of . The algorithm induces a classifier from, and estimates its accuracy rate with.

where if and 0 otherwise. The accuracy estimation is the overall number of correct classifications, divided by the number of samples in the dataset. Thus, the cross-validation accuracy estimation depends on the division into folds. The complete cross-validation is the average of all possibilities for choosing samples out of, but the computational cost is usually too expensive. In our experiments, we take a single split of the data into folds in place of complete *k*-fold cross-validation.

III. RALATED WORK

In supervised classification, each classifier has an associated prediction error. The prediction error is the probability of wrong classification of unlabeled instances. Naonori and Ryohei had proved the minimum theoretical prediction error is the Bayes error, which given by the Bayes classifier [3]. Since the Bayes classifier depends only on the feature-label probability distribution of the domain, the Bayes error does not depend on training data or sample size. Any other classifier has a higher than or equal error as the Bayes classifier.

In most real-world problems, the prediction error cannot be exactly calculated, and must be estimated from data, but available data set is generally rather small. For -NN classifiers, the weighted rule is the strategies to resolve the ties in limited sample set. Dudani proposed that when the number of training samples is small or of moderate size, then the distance-weighted rule will yield a smaller probability of error than the majority rule [9]. In 1978, T.BAILEY and A.K. JAIN duplicated the experiments reported by Dudani, and proved that the majority rule had the lowest probability of error among all distance-weighted rules for moderate values of () in the infinite sample case [10].

It is generally accepted that the error estimation of a classifier has higher variance and lower bias as the number of required parameters increases, or equivalently, as the sensitivity to the changes in the training sets increases [11,12]. In -fold cross validation, the data set is divided into folds. Finally, the -fold cross validation estimation of the error is the average value of the errors committed in each fold. Thus, the -fold cross validation error estimation depends on two factors: the training set and the partition into fold. So the estimated error can be considered a random variable which depends on the training set and the partitions. Meanwhile, Juan Diego Rodríguez et al [2] analyzed the variance of the *k*-cv considering its sources of variance: sensitivity to changes in the training set and sensitivity to changes in the folds. Ron Kohavi has demonstrated that if the induction algorithm is stable for a given datasets, the variance of the cross validation estimates should be approximately the same, independent of the number of folds [1].

IV. EXPERIMENTAL STUDY

A. Accuracy

This section describes the experimental studies performed. Considering the computational complexity in experiments, firstly, we define in *k*-nearest neighbors classification algorithms. Because previous works demonstrate that the value is a relatively small integer, there is little advantage in choosing larger than 3. We implement experiments on the attribute-weighted *k*-NN classifier and the instance-weighted *k*-NN classifier, the simplified pseudo-code is as follows.

```
Algorithm: weighted k-NN classifier

read datafile(training data)
read datafile(testing data)

InitialAttWeights()
{ Initialize the weights with random values.
}
CrossValidation()
{learn weights by cross validation (3 fold) on training set.
}
LearnWeight()
```

```
{ learn weights on the whole training set
}
for each testing pattern do
ClosestNeighbors(k)
{Find k nearest neighbors of every test pattern in training set.
L={}
for (j=0,j<NO_OF_ATT-1,j++)
if(attribute-weighted)
Distance+=
else
Distance+=
L={Closest(sqrt(Distance))
}
if(attribute-weighted)
testClass=Major(ClosestNeighbors.classvalue)
else
{
tmpWeight[classvalue]=
SUM(ClosestNeighbors[classvalue].weight)
If(max(tmpWeight[classvalue]))
testClass=classvalue
}
accuracy=(correctClassifiedInstance/testInstancesNum)
```

Recent theoretical and experimental results have shown that it is not always beneficial of increasing the computational cost, especially if the relative accuracies are more important than the exact value.

In this section, we perform experiments with three different real-world datasets. The properties of these datasets are given in Table I. Numbers of attributes includes the class attribute which is the last attribute, and the class value should be integers only like 0,1,2. All the data-sets have only numeric valued attributes.

Table II compares classification accuracies obtained from *k*NN classifier, attribute-weighted *k*NN classifier and instance-weighted *k*NN classifier with 3-fold cross-validation.

Then we change the *k* value of *k*-nearest neighbors from 3 to 15 in attribute-weighted *k*NN classifier and instance-weighted *k*NN classifier on a given dataset (HEART), and compare the accuracy of attribute-weighted *k*NN classifier with or without 3-fold cross validation which is for adjusting the weights, the results are presented in Figure 1 and Figure 2. Considering the computational complexity, we define for *k*-cv in this process even if 10-cv is less bias than.

TABLE I PROPERTIES OF THE DATA-SETS USED

Data-set	Number of attributes	Number of classes	Number of training instances	Number of test instances
WINE	14	3	146	32
HEART	14	2	224	46
HILL VALLEY	101	2	606	606

TABLE II. CLASSIFICATION ACCURACY COMPARISON IN VARIOUS DATASETS (WITH 3-FOLD CROSS-VALIDATION)

Data-set	*k*NN	Attribute-Weighted *K*nn Classifier	Instance-Weighted *K*nn
WINE	87.5%	87.5%	50%
HEART	78.2609%	80.4348%	56.6956%
HILL VALLEY	50.9901%	48.0198%	50.33%

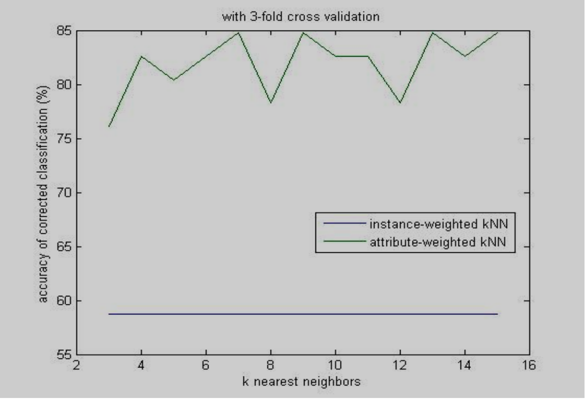
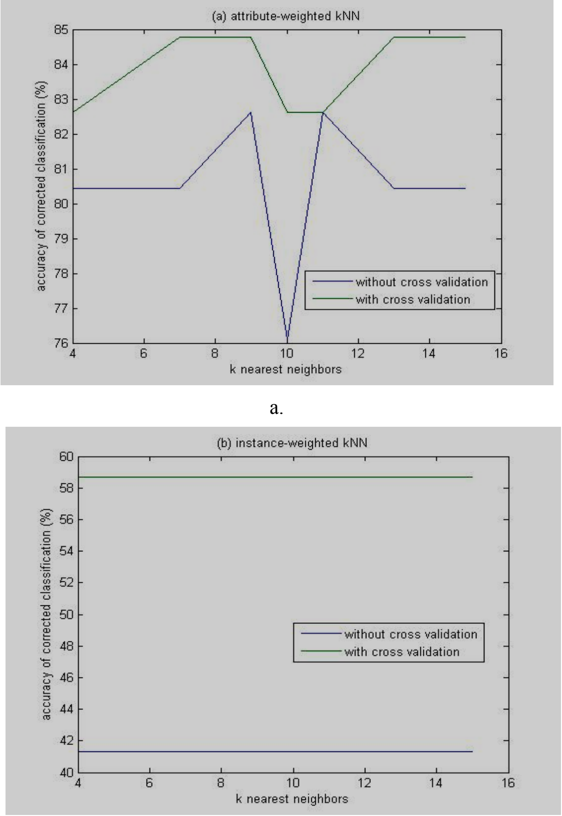


Figure 1 While the k value change from 3 to 15, the classification accuracy of instance-weighted kNN algorithm and attribute-weighted kNN algorithm are affected differently on a given dataset.



a.

b.

Figure 2 (a) and (b) show that the cross validation is good for improving classification accuracy by adjusting the weights.

B. Stability

An inducer is stable for a given dataset, that is to say, it induces classifiers which make the same predictions when it is given the perturbed datasets. In this section, we empirically study the statistical properties of the -fold cross validation estimator using different parameters. In weighted *k*NN, the weights are adjusted in training phase until the cross-validation estimate of accuracy is desired.

On the given dataset (HEART), we expect the perturbation caused by deleting the instances for the folds in *k*-fold cross validation, the cross validation estimate will be unbiased. Then we run attribute-weighted *k*NN classifier and instance-weighted *k*NN classifier with *k*-fold cross validation for various *k* value while in *k* nearest neighbors. The results are presented in Figure 3.

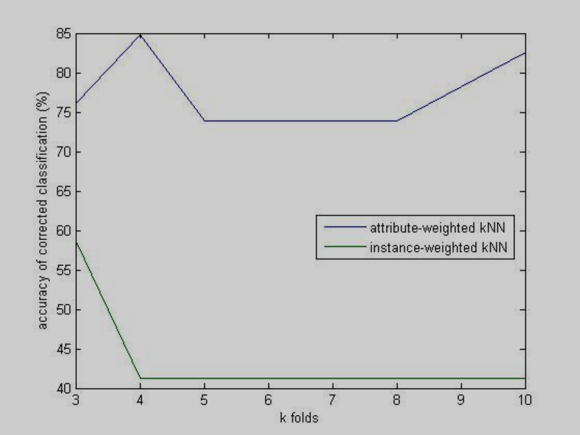


Figure 3 On a given dataset, the size of every fold decrease when the number of folds increase. The experimental results show that instance-weighted kNN is relatively stable.

V. RESULTS and DISCUSSION

We now show the experimental results and discuss their significance.

We respectively change the training samples and the test samples to verify the sensitivity of the 3-fold cross validation estimator in different weighted *k*NN classifiers. From the presented results in TABLE II, it is observed that the accuracy rate of weighted *k*NN classifier is not always higher than simple *k*NN classifier on a given dataset, just only the attribute-weighted *k*NN classifier is better than the other proposed classifiers when the dataset is low dimensionality. But when the dataset becomes high dimensionality (such as more than 100 attributes), the performance of these three classifiers is dramatically reduced, that is to say the “Curse Dimensionality” is caused. Meanwhile, the performance of instance-weighted *k*NN classifier is relatively stable, even though the size of dataset and the number of attribute become large. It is indicated that the instance-weighted *k*NN classifier is relatively insensitive to the changes in the training sets than other two *k*NN classifiers.

The experimental results presented in Figure 1 show that the accuracy rate is stable no matter how the nearest neighbors value changes because one instance has same weight. The accuracy rate of attribute-weighted classification algorithm fluctuates because every attribute of one instance has different weight produced by random and adjusted by cross validation. The results presented in Figure 2 show that the cross validation is beneficial for improving the classification accuracy on a given dataset. Also that result exactly verifies all available samples must be split into training and test set, which is same to D.J.Hand’s view.

Ron Kohavi [1] has demonstrated that if the inducer is almost stable under the perturbations caused by deleting the instances for the folds in *k*-fold cross validation on a given dataset, the accuracy estimation is likely to be reliable. That is to say, the variance of *k*-fold cross validation does not depend on *k*. From Figure 3, we can conclude that instance-weighted *k*NN classifier more relatively stable than attribute-weighted *k*NN classifier, there is almost no change in the variance of the cross validation estimate when the number of folds is varied while the training set and the test set is independent.

VI. SUMMARY AND CONCLUSIONS

We illustrated two weighted functions for improving *k*NN classifiers, and reviewed common accuracy estimation methods, especially for *k*-fold cross validation. We have compared the two weighted approaches on a variety of real-world datasets with differing parameters. Experiment 1 shows that accuracy of weighted classifiers. Experiment 2 indicated that if the induction algorithm is relatively stable for a given dataset, the classification accuracy should be approximately the same, independent of the number of folds. But in reality, a complex inducer is unlikely to be stable for large perturbations, unless it has reached its maximal learning capacity. For improving pattern classification accuracy in lower dimensionality, we recommend using attribute-weighted *k*NN and *k*-fold cross validation for adjusting weight, while the number of fold is no more than ten. For model selection, there is no the best classifier suitable for all domains, each classifier has its own advantage in some domain.

ACKNOWLEDGMENT

This paper is funded by Henan Science and Technology Agency of Soft Science Project, Project Number 152400410344.

REFERENCES

[1] Ron Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. IJCAI.(1995) .

[2] Juan Diego Rodríguez; Aritz Pérez; Jose Antonio Lozano, Sensitivity Analysis of k-Fold Cross Validation in Prediction Error

Estimation, IEEE Transaction on pattern analysis and machine intelligence, vol.32,no.3, 569-575 (2010).

[3] Naonori UEDA; Ryohei NAKANO, Estimating Expected Error Rates of Neural Network Classifiers in Small Sample Size Situations:A comparison of Cross-Validation and Bootstrap. Neural Networks, 1995. Proceedings., IEEE International Conference on, vol.1, no., 101-104(1995).

[4] MAYANKA B. KHUMAN, Classification of Remote Sensing Data Using K-NN Method. Journal of Information, Knowledge and Research in Electronics and Communication Engineering, vol.02, No.02,817-821(2012).

[5] D.J.Hand, Recent Advanced in Error Rate Estimate. Pattern Recognition Letters,vol.4,no.5, 335-346(1996).

[6] Anil K.Jain; Robert P.W.Duin; Jianchang Mao. Statistical Pattern Recognition: A Review. IEEE Transactions on pattern analysis and machine intelligent, vol. 22, No.1, 4-37(2000).

[7] Efron,B. &Tibshirani, R.(1993), An introduction to the bootstrap, Chapman & Hall.

[8] Sanjeev R. Kulkarni; Gábor Lugosi; Santosh S. Venkatesh, Learning Pattern Classification-A Survey. IEEE Transactions on Information Thoery,vol.44,no.6,2178-2206(1998). M.Stone, Cross-Validatory Choice and Assessment of Statistical Predictions. Journal of Royal Statistical Soc.Series B, vol.36, 111-147(1974).

[9] S.A.Dudani, The distance-weighted k-nearest neighbor rule. IEEE Trans.Syst.,Man,Cybern.,vol.SMC-6, 325-327(1976).

[10] T.BAILEY; A.K. JAIN, A Note on Distance-Weighted k-Nearest Neighbor Rules. Systems, Man and Cybernetics, IEEE Transactions on, vol.8, no.4, 311-313(1978).

[11] C.M. Bishop, Pattern Recognition and Machine Learning. Spring, (2006).

[12] R.O. Duba; P.E. Hart; D.G. stork, Pattern Classification. second ed. Wiley Interscience(2000).

[13] Michael Kearns,A Bound on the Error of Cross Validation Using the Approximation and Estimation Rates, with Consequences for the Training-Test Split. Neural Computation, vol.,no.9,1143-1161(1997).

[14] Atashpaz-Gargari, E.; Chao Sima; Braga-Neto, U.M.; Dougherty, E.R., Relationship between the accuracy of classifier error estimation and distribution complexity. Genomic Signal Processing and Statistics (GENSIPS), 2011 IEEE International Workshop on , vol., no., 147-149(2011).

[15] Viswanath, P.; Sarma, T.H., "An improvement to k-nearest neighbor classifier," in Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE , vol., no., pp.227-231, 22-24 Sept. 2011, doi: 10.1109/RAICS.2011.6069307.