

Research Article

Extensions of k-Nearest Neighbor Algorithm

Rashmi Agrawal

Faculty of Computer Applications, Manav Rachna International University, Faridabad,
Haryana 121004, India

Abstract: The aim of this study is to review various extensions of the nearest neighbor algorithm and discuss their approach along with limitations of the method. In nonparametric classification, no prior information is required for predicting the class label. k-Nearest Neighbor is the simplest and well known algorithm used in data mining. The various extensions of k-nearest neighbor algorithm which have been studied are weighted nearest neighbor, feature selection methods, fuzzy nearest neighbor, genetic algorithm based classifiers and nearest neighbor algorithm using ensembling techniques.

Keywords: Ensembling, feature selection, fuzzy set, genetic algorithm, nonparametric classification, weighted nearest neighbor

INTRODUCTION

Classification is the process of assigning objects to some predefined categories. This technique encircles many assorted applications like spam email detection, based on result of MRI scans categorization of cells as malignant or benign, prediction of the customers as loyal or defaulter in loans etc.

A classification technique is an organized approach for building classification model from the given input dataset. Some of the well-known classification techniques are neural networks by Haykin (1998), rule-based classifiers by Liu *et al.* (2000), k-Nearest Neighbor classifier, decision tree (survey given by Safavian and Landgrebe (1991) and naïve bayes classifier (empirical study given by Rish (2001). The learning algorithm of each technique is employed to build a model which is used to find the relationship between the attribute set and class label of the given input data. The key objective of the technique is to build a model with best generalization fitness. Decision trees and rule-based classifiers are examples of eager learners or active learners because their model maps the input attributes to the class labels immediately in the training data. On the other hand, some classifiers delay this process until it is needed to classify. They are known as lazy learners. K-Nearest Neighbor classifier, a nonparametric classifier, is the best example of a lazy learner.

The nearest neighbor method given by Fix and Hodges (1951) is one of the simplest techniques in predictive mining. Let x be an input with j features (a_{i1} ,

a_{i2}, \dots, a_{ij}) and n be the total number of inputs, then the distance between input x and x_j ($j = 1, 2, \dots, n$) is defined using Euclidean distance measure as:

$$d(x, x_j) = \sqrt{\sum_{j=1}^n (x - x_j)^2}$$

The class label for the input point is the class label of the nearest neighbor. The first extension to this idea is the k-nearest neighbor, which is widely used and well known algorithm. Here not only the nearest neighbor but k nearest neighbors are considered for predicting the final class label. This parameter k is given by the user. The classification performance varies significantly with different values of k . This value of k defines the space of the neighborhood. If k is taken as 1, then class label is determined by the class label of nearest neighbor. When k is equal to number of samples, this information becomes general. Typically large value of k may increase the classification accuracy but distribution of neighbors may belong to more than one class which may result into incorrect class label.

Extensions of the K-nearest neighbor classifier:

Weighted nearest neighbor techniques: In weighted KNN classifier, the classification process consists of the following steps:

- Assign each neighbor x_i a weight w_i using a weight assignment technique.
- Assign class label q_c to the query point using the rule given below:

$$q_c = \min w_j (\sum_1^n w_i)$$

The simplest approach to find the class label is assigning the majority class among the nearest neighbor. It makes sense to add weights to the neighbors and the general technique is to find the weight of a class by inverse of their distance to the query (q):

$$W(c_j) = \sum \frac{1}{d(q, x_i)^n} 1.(c_j, c_i)$$

where, c_j is the class label and $d(q, x_i)$ is the distance between query point and test example and (c_j, c_i) returns 1 if class label match otherwise 0.

Shepard (1987) gave another approach for weight which uses an exponential function instead of inverse:

$$w(c_j) = \sum_{i=1}^k e^{-\frac{d(q, x_i)}{k}} 1.(c_j, c_i)$$

As an improvement to KNN, Dudani (1976) introduced distance-weighted KNN (WKNN) algorithm. However, WKNN does not produce satisfactory results due to the existence of outliers, particularly in small sample size dataset.

On the basis of WKNN by Dudani (1976), a new Distance-Weighted K-Nearest Neighbor Rule (DWKNN) was given by Gou *et al.* (2012) using dual distance-weighted function. They employed the dual distance-weights neighbors to find out the class of the object. Simply majority voting for KNN may not be effective if the neighbors vary widely with their distances. In DWKNN the original weight is multiplied with a new weight to determine the dual weight. This method reduces the weight of nearest neighbors and provides too much weight to the outliers as compared to the WKNN and improving the classification performance. However, DWKNN is not effective with irregular class distribution.

Zuo *et al.* (2008) defined the weighted KNN rule as a constrained optimization problem and contributed a kernel difference weighted k- nearest neighbor (KDF-KNN). Unlike DS-WKNN given by Dudani (1976), KDF-KNN first finds out the k nearest neighbors and then evaluates the difference between nearest neighbors and the query point. They took the weight assignment as constraint optimization problem, formulated as quadratic programming problem, which can be solved using Langrangian Multiplier method. The KDF-KNN method was applied to 30 standard data sets and results obtained were better than KNN and DWKNN.

Hechenbichler and Schliep (2004) gave the idea that the observations close to new observation should get higher weight as compared to other neighbors. To achieve this, nearest neighbors have to be transformed into similarity measures which can be further used as

weights. After finding the similarity measure for the training dataset, each input is classified into the class with largest added weight. This method cannot solve the problem of variable selection. Too many covariates vary completely at random and thus do not have any prediction for target and as such may cause severe disturbance to the prediction.

Feature selection techniques: Identification of relevant features and removal of other irrelevant features has been an interesting problem in the area of machine learning. In Langley (1994) reviewed this problem and described it in the form of heuristic search. He presented the task of feature selection as a search problem having a subset of possible features in each state. The four issues were addressed:

- To determine forward selection or backward selection
- Adding and removing features at each decision point
- Strategy to evaluate alternative subset of attributes (filter method or wrapper method)
- Criteria for halting the search

Feature selection is of extreme importance to enhance the speed of learning and to improve the quality. Kira and Rendell (1992) presented a new feature selection algorithm RELIEF which uses a statistical method and does not include heuristic search. This algorithm takes the assumption that scale of every feature is either nominal or numerical. A function is defined to update the feature weight vector for every sample and the average feature weight called 'Relevance' is determined. However, this algorithm is valid only when the relevance is large for relevant features and small for other features.

Marchiori (2013) investigated a decomposition of RELIEF into the class dependent feature weight terms. They showed that adding complementary characteristics of a feature in different classes neutralize each other otherwise they may give different weight contributions. Therefore, relevance of some features for a single class may not be detected.

In feature selection techniques, generally a search strategy is incorporated to explore the space of subsets of features which includes methods for finding starting point and generating candidate subsets and evaluation criteria to compare the suitability of candidates. This evaluation scheme can be divided into two broad categories:

- **Filter approach:** In this approach the irrelevant features are removed from the set of features before applying the learning objective.
- **Wrapper method:** In this method the learning algorithm is used to select the features from the feature set.

The limitation of filter approach is that features are considered in isolation. Therefore two strongly correlated features either may be ignored or may be redundant. Wrapper method overcomes the limitations of filter approach as classifier is itself wrapped in feature selection process. It is done either through forward selection or through backward selection. The forward selection starts with no features and each feature is added at a step. The backward selection process starts by considering all features initially and irrelevant features are removed at every step.

Das (2001) examined the pros and cons of filter and wrapper methods used in feature selection and proposed a new hybrid algorithm. This algorithm was based on the concept of boosting from computational learning theory. He presented Boosted Decision Stumps for Feature Selection (BDSFS) algorithm by using Adaboost to bridge the gap and giving more informed filter method. The algorithm used boosted decision stumps as the weak learners. The algorithm was, however, not well suited for multi-class datasets.

Fuzzy K-nearest neighbor techniques: In the KNN classifier each of the samples in the training set is given equal importance for the assignment of class label. This may cause problems at the places where sample sets overlap. Also there is no indication of the membership of the input sample to a class. Keller *et al.* (1985) addressed this issue by incorporating fuzzy-set theory in the KNN. Fuzzy sets were introduced by Zadeh (1965) and derived by generalizing the concept of a characteristic function to a membership function. A fuzzy set gives the advantages of degree of membership in a set rather than binary value of belongingness. A fuzzy technique specifies the degree to which an object belongs to each class. In the fuzzy k-nearest neighbor algorithm class membership is assigned to a sample vectors rather than assigning any vector of particular class. The algorithm was tested on three datasets and based on the membership, samples were correctly classified.

Based on the fuzzy set theory, Bian and Mazlack (2003) contributed fuzzy-rough nearest neighbor approach which is more suitable for unbalanced data. They incorporated rough uncertainty into the fuzzy KNN classifier and named it as fuzzy-rough nearest neighbor classification approach. It is generalization of the fuzzy nearest neighbor approach given by Keller. The rough set method was introduced by Pawlak (1998). The rough set theory is based on the assumption that for every object in the universe there is some associated crisp concept or information that can approximate it. Objects characterized by same set of information are indiscernible. The mathematical basis of rough set theory is the indiscernibility relation generated in this way. The fuzzy-rough nearest neighbor approach contains upper as well as lower membership degree and hence more meaningful

interpretations can be drawn. Unfortunately, this study also has certain limitations. However, the approach was limited for small datasets and was not able to handle the datasets with missing values.

The fuzzy KNN algorithm collects the evidences from k-closest neighbors and hence it is difficult to choose proper value of k. Fuzzy-rough version of KNN proposed by Bian and Mazlack (2003) is not suitable when training pattern is noisy. In Sarkar (2007) investigated fuzzy and rough uncertainties in KNN to overcome the drawbacks of conventional KNN. They observed that class labeling associates with two types of uncertainties i) The fuzzy uncertainty which is due to the overlapping of classes and ii) Rough uncertainty which is due to insufficient features. To model this, Sarkar (2007) employed the concept of fuzzy-rough set in KNN and Fuzzy-Rough Nearest (FRNN) algorithm was proposed. The algorithm was built in a manner so that its output was interpreted as fuzzy-rough ownership. FRNN considers all training patterns as neighbors with different degree unlike considering K nearest neighbors in conventional KNN. Hence it avoids the problem of selecting optimum value of k. However, the FRNN has certain drawbacks and not considered so good in terms of space and time complexity.

Genetic algorithm: Genetic Algorithms were first developed by Goldberg and Holland (1988), are well known computational models today. Genetic algorithms have been an effective tool in data mining and are useful when mathematical analysis does not exist to narrow down the search space for large and complex solutions. Genetic algorithms ponder that solution to the given problem can be found in the genetic pool of the population but is possible only after the association of different genes using genetic operations. These algorithms have been successful in various data mining techniques such as association rule mining, classification and clustering. Genetic algorithm can be applied in data mining in two ways; one using hypothesis testing and refinement when some hypothesis is presented by user and system evaluates hypothesis and then refines it. Second is to design some hybrid techniques by blending known data mining techniques with genetic algorithms.

Kelly and Davis (1991) used genetic algorithm to enhance the performance of KNN classifier. They observed that when attributes are misleading or irrelevant to classification the conventional KNN may be less effective. Therefore to make the distance measure more meaningful, a real valued genetic algorithm is needed to find vectors of attribute weighting. The genetic algorithm weighted KNN (GA-WKNN) algorithm combines the optimization capabilities of genetic algorithm and weighted KNN algorithm. GA-WKNN was tested on three datasets and cross validation error estimation techniques were

performed. It was observed that the GA-WKNN algorithm is an improvement over KNN but a number of issues such as class based derivation of attribute weight vector, techniques for learning real valued weight vectors and choice of optimum k remain unaddressed.

Ho *et al.* (2002) provided a novel Intelligent Genetic Algorithm (IGA) which is superior to the conventional genetic algorithms in solving optimization problems. They developed a method for designing an optimal 1- nearest neighbor classifier using intelligent genetic algorithm with an intelligent genetic algorithm with an intelligent crossover operation of genetic algorithm. This algorithm is successful to solve large parameter optimization problem but is limited to the one nearest neighbor and does not considers other neighbors.

Aci *et al.* (2010) contributed a hybrid method by integrating Bayesian method and Genetic algorithm in the KNN classifier. They applied the Bayesian method based expectation maximization on the selected data set and then k-nearest neighbor method was applied. Genetic algorithm was applied on the last generated dataset. Data were sorted repeatedly according to their distances to define the crossover ratio. The method was tested on breast cancer, Iris, glass, yeast and wine datasets and showed improvements as compared to conventional methods. However, this method is useful only for the small datasets.

Ensembling techniques: Ensembling techniques are an active area of research in supervised learning. In machine learning, an individual model may behave as an expert opinion for a particular dataset and may provide the best results whereas it may not produce appropriate results for the other problem. In such cases, to obtain more reliable decisions, different models can be combined. It has been found by the researchers that ensembles are more accurate than an individual classifier. Recently Ensemble methods have been most influential methods in machine learning. Multiple models are combined into one which is more accurate than any of its components. Two main algorithms for Ensembling are Bagging and Boosting.

Bagging or Bootstrap Aggregation was introduced by Brieman. It is an ensemble method for improving accuracy of a classifier. It was given as a variance reduction technique by Breiman (1996). Due to its simplicity and easiness of implementation it has taken much attention by the researcher. Büchlmann and Yu (2002) showed bagging techniques to improve the accuracy of classification trees. Büchlmann and Yu (2002) showed bagging as a smoothing operation and this amount of smoothing is generated automatically. They discussed those Bagging smoothes indicator functions, which in some of their base procedures are inherent. They also discovered that due to averaging over different predictor variables, bagging may also

have a positive effect. The main disadvantage of Bagging is lack of interpretation.

Boosting is another principled approach of Ensembling. In this iterative approach, misclassified instances are re-weighted and their importance is increased for resulting models, thus increasing chance for fixing the errors. A famous algorithm for boosting is AdaBoost also known as Adaptive Boosting given by Freund and Schapire (1996). Bagging is a parallel ensemble method whereas boosting methods are sequential.

In AdaBoost, in the training phase a sequence of classifiers is produced using same base classifier. Each classifier is dependent on the previous classifier and it focuses on the errors of previous classifier. In each iteration, incorrectly predicted examples are given higher weights and correctly predicted examples are given lower weights. In the testing phase, the results of sequence of classifiers are combined to determine the final outcome. AdaBoost algorithm is formulated for binary class problem and not easily applied on multiclass problems.

Amores *et al.* (2006) used functional approximation to estimate the similarity function as generalization method instead of estimating a parametric or similarity distance. For this the AdaBoost with decision stumps is used. This method has various advantages. The proposed distance estimation method is applicable for all kind of classifiers and is not based on parametric approach. The method is effective when training set is small and the estimated similarity uses a small number of dimensions resulting in dimensionality reduction. However, this is only applicable for small training set.

Athitsos and Sclaroff (2005) contributed a method to combine boosting with k- Nearest Neighbor classifier. They took a large number of distance measures as input and weighted linear combinations of these distance measures as outputs. For achieving the accuracy these output distance measures were optimized. The algorithm was applied on eight pattern recognition datasets and gave lower error rates as compared to AdaBoost algorithm. This algorithm converts multi-class problem to a single binary class problem but hard to apply where number of classes are more.

The boosting algorithms work fine for two class problems but it is difficult and complex to predict the class label for multiclass problems.

Data uncertainty: The existence of uncertainty in data disturbs the results of data mining techniques. The features which have a greater level of uncertainty need to be cured in a different way as paralleled to the features having poorer level of uncertainty details given by Agrawal (2014). The basic KNN method is unable to handle the uncertainty and imprecision in the labeling of known classes. This can lead to a problem as in real

life in many cases uncertainty arises. Agrawal and Ram (2015a) reviewed various existing data classification techniques for uncertain data using the k nearest neighbor approach. Agrawal and Ram (2015b) proposed a new effective distance measure to handle the features having uncertain attributes. This method is however able to handle only the numerical features with uncertain values.

CONCLUSION

Although the extensions of k-Nearest Neighbor classifiers discussed so far has shown remarkable improvement in the accuracy of classifier as compared to the traditional method but there are some limitations of each algorithm. In future the researchers should focus on designing and developing a classification algorithm using K-Nearest Neighbor to remove the deficiencies pointed out above and to produce the results with improved accuracy.

ACKNOWLEDGMENT

My thanks are due to my supervisor Dr Babu Ram, Professor, Manav Rachna International University, Faridabad for his constant guidance and support.

REFERENCES

- Aci, M., C. İnan and M. Avci, 2010. A hybrid classification method of k nearest neighbor, Bayesian methods and genetic algorithm. *Expert Syst. Appl.*, 37(7): 5061-5067.
- Agrawal, R., 2014. K-nearest neighbor for uncertain data. *Int. J. Comput. Appl.*, 105(11): 13-16.
- Agrawal, R. and B. Ram, 2015a. A modified K-nearest neighbor algorithm to handle uncertain data. *Proceeding of the 5th International Conference on IT Convergence and Security (ICITCS, 2015)*, pp: 1-4.
- Agrawal, R. and B. Ram, 2015b. A survey of uncertain data mining techniques. *Int. J. Adv. Res. Educ. Technol. Manage.*, 3(1): 252-256.
- Amores, J., N. Sebe and P. Radeva, 2006. Boosting the distance estimation: Application to the k-nearest neighbor classifier. *Pattern Recogn. Lett.*, 27(3): 201-209.
- Athitsos, V. and S. Sclaroff, 2005. Boosting nearest neighbor classifiers for multiclass recognition. *Proceeding of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pp: 45.
- Bian, H. and L. Mazlack, 2003. Fuzzy-rough nearest-neighbor classification approach. *Proceeding of the 22nd IEEE International Conference of the North American on Fuzzy Information Processing Society (NAFIPS, 2003)*, pp: 500-505.
- Breiman, L., 1996. Bagging predictors. *Mach. Learn.*, 24: 123-140.
- Büchlmann, P. and B. Yu, 2002. Analyzing bagging. *Ann. Stat.*, 30(4): 927-961.
- Das, S., 2001. Filters, wrappers and a boosting-based hybrid for feature selection. *Proceedings of the 18th International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., pp: 74-81.
- Dudani, S.A., 1976. The distance-weighted k-nearest-neighbor rule. *IEEE T. Syst. Man Cyb.*, SMC-6(4): 325-327.
- Fix, E. and J.L. Hodges, 1951. Discriminatory analysis, nonparametric discrimination: Consistency properties. *Technical Report 4*, U.S. Air Force, School of Aviation Medicine, Randolph Field, TX.
- Freund, Y. and R.E. Schapire, 1996. Experiments with a new boosting algorithm. *Proceeding of the 13th International Conference on Machine Learning*, pp: 1-9.
- Goldberg, D.E. and J.H. Holland, 1988. Genetic algorithms and machine learning. *Mach. Learn.*, 3(2): 95-99.
- Gou, J., L. Du, Y. Zhang and T. Xiong, 2012. A new distance-weighted k -nearest neighbor classifier. *J. Inform. Comput. Sci.*, 9(6): 1429-1436.
- Haykin, S.S., 1998. *Neural Networks: A Comprehensive Foundation*. 2nd Edn., Prentice Hall, Englewood Cliffs, NJ, pp: 696.
- Hechenbichler, K. and K.P. Schliep, 2004. Weighted k-nearest-neighbor Techniques and Ordinal Classification. *Discussion Paper 399*, SFB 386, Ludwig-Maximilians University, Munich.
- Ho, S.Y., C.C. Liu, S. Liu and J.W. Jou, 2002. Design of an optimal nearest neighbor classifier using an intelligent genetic algorithm. *Proceeding of the Congress on Evolutionary Computation (CEC '02)*. Honolulu, HI, 1: 594-599.
- Keller, J.M., M.R. Gray and J.A. Givens, 1985. A fuzzy k-nearest neighbor algorithm. *IEEE T. Syst. Man Cyb.*, SMC-15(4): 580-585.
- Kelly, J.D. and L. Davis, 1991. A hybrid genetic algorithm for classification. *Proceeding of the 12th International Joint Conference on Artificial Intelligence (IJCAI'91)*, 2: 645-650.
- Kira, K. and L.A. Rendell, 1992. A practical approach to feature selection. *Proceeding of the 9th International Workshop on Machine Learning*, pp: 249-256.
- Langley, P., 1994. Selection of relevant features in machine learning. *Proceeding of the AAAI Fall Symposium on Relevance*. New Orleans, LA, pp: 140-144.
- Liu, B., Y. Ma and C.K. Wong, 2000. Improving an association rule based classifier. *Proceeding of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD '00)*, pp: 504-509.

- Marchiori, E., 2013. Class dependent feature weighting and k-nearest neighbor classification. Proceeding of the 8th IAPR international conference on Pattern Recognition in Bioinformatics (PRIB'13), pp: 69-78.
- Pawlak, Z., 1998. Rough set theory and its applications to data analysis. *Cybernet. Syst. Int. J.*, 29(7): 661-688.
- Rish, I., 2001. An empirical study of the naive Bayes classifier. Proceeding of IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, IBM, New York, 3(22): 41-46.
- Safavian, S.R. and D. Landgrebe, 1991. A survey of decision tree classifier methodology. *IEEE T. Syst. Man Cyb.*, 21(3): 660-674.
- Sarkar, M., 2007. Fuzzy-rough nearest neighbor algorithms in classification. *Fuzzy Set. Syst.*, 158(19): 2134-2152.
- Shepard, R.N., 1987. Toward a universal law of generalization for psychological science. *Science*, 237(4820): 1317-1323.
- Zadeh, L.A., 1965. Fuzzy sets. *Inform. Control*, 8(3): 338-353.
- Zuo, W., D. Zhang and K. Wang, 2008. On kernel difference-weighted *k*-nearest neighbor classification. *Pattern Anal. Appl.*, 11(3): 247-257.