ISSN:2320-0790

An International Journal of Advanced Computer Technology

Combined Approach for Improving Accuracy of Prototype Selection for k-NN Classifier

Shikha Gadodiya[1], Manoj B. Chandak[2]

[1]M.Tech Student, CSE Department, Shri Ramdeobaba College of Engineering and Management, Nagpur, India
[2]Professor, CSE Department, Shri Ramdeobaba College of Engineering and Management, Nagpur, India

Abstract: The k-nearest-neighbour classifier is a powerful tool for multiclass classification and thus widely used in data mining techniques. But it consists of some severe drawbacks: high storage requirements, low noise tolerance and low efficiency in classification response. The solution to these drawbacks is to apply nearest neighbor on the reduced dataset which can be obtained by applying Prototype Selection methods on original training dataset. Various Prototype Selection methods have been developed yet but are not that efficient to overcome all the drawbacks simultaneously. So here is an attempt to build relatively more efficient algorithm by combining two or three previously developed approaches.

Keywords: k-nearest neighbor classifier, data reduction, prototype selection, efficiency.

I. INTRODUCTION

Though k-NN classifier has been used widely as a classification technique in various data mining tasks, it is not that efficient because of its main drawbacks of storage space, calculation cost and noise tolerance. So the Prototype Selection approach has been introduced a decade ago to overcome these drawbacks. Prototype Selection is usually viewed as purely a means toward building an efficient classifier. Prototype Selection methods seek a minimal subset of samples that can serve as a condensed view of a data set. As the size of modern data sets grows, and k-NN suffers from high storage space requirement, being able to present a domain specialist with a short list of "representative" samples chosen from the original data set is of increasing interpretative value.

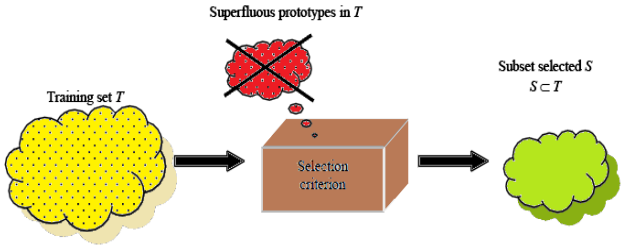

Fig. 1: Prototype Selection Mechanism

For classifying new prototypes a training set is used which provides information to the classifiers during the training stage. In practice, not all information in a training set is useful therefore it is possible to discard some irrelevant prototypes. This process is known as “prototype selection”. There are 50+ such methods proposed yet depending on the qualities of the families which they belong to. These methods are capable of overcoming the drawbacks of k-NN classifier but not all drawbacks simultaneously. In this thesis there is an attempt to propose a method by combining two previously invented methods to overcome all 3 drawbacks viz. high storage requirement, low classification efficiency and low noise tolerance simultaneously.

Here we have made the use of KEEL (Knowledge Extraction Based on Evolutionary Learning) data mining software tool to analyse the results of our approach. It is an open source java based software that supports data management and a designing of experiments, specially the implementation of evolutionary learning and soft computing based techniques for Data Mining problems including regression, classification, clustering, pattern matching and so on.

II. RELATED WORK

Prototype Selection methods proposed so far belongs to particular families according to their qualities. These families of prototype methods are classified depending upon the taxonomy that it follows. The properties involved in the definition of taxonomy are as follows:

A. ***Order of Search:***

- Incremental process
- Decremental process
- Batch process
- Mixed process
- Fixed process

B. ***Type of Selection:***

- Condensation approach
- Edition approach
- Hybrid approach

C. ***Evaluation of Search:***

- Filter class
- Wrapper class

Depending on this taxonomy and properties we have selected two possibly best methods from better family to create the combined approach from them. First, is DROP (Decremental Reduction Optimization Procedure) method belonging to the Decremental-Filter family. Second, is CPruner method again belonging to Decremental-Filter family. The details of these previously developed methods are as follows:

DROP Algorithm: Drop was designed taking into account the effect of the order of removal on the performance of the algorithm that is, it is insensitive to the order of presentation of the instances. In this, the instances are ordered by the distance to their nearest neighbor. The instances are removed beginning with the instances furthest from its nearest neighbor. This tends to remove the instances furthest from the boundaries first.

i) CPruner Algorithm: CPruner first tends to identify the pruned instances among your original dataset and then they can be removed. The instances can be pruned when they satisfy following two conditions: It should be noisy instance and it should be superfluous instance but not a critical one. Then with certain rule of order of instance removal, the pruned instances are removed from dataset.

But these two algorithms were not found much efficient according to the experiment performed on KEEL tool and result obtained.

III. EXPERIMENT PERFORMED

The above two mentioned algorithms were tested on the large data set on viz. Marriage Dissolution in the USA which was adapted from an example in the software package aML, and is based on a longitudinal survey conducted in the U.S. It is available as data mining dataset in open source. The KEEL tool's inbuilt methods DROP3 and CPruner were applied on this dataset. The KEEL gui, experimental setup and statistical results of both the methods found from this experiment are as shown in following figures. The results found for given dataset with additional 10% noise were not that satisfactory. For more efficient data mining tasks, we need more accurately classified dataset. The present methods are on an average 0.73 units accurate, but if the data set becomes more noisy then its accuracy decreases. Here Drop3 which works more on classification of data proved to be 0.76 units accurate and CPruner which mainly works on noise reduction proved to be 0.70 units accurate on classification task. But, again this accuracy rate decreases when the dataset size keeps on increasing and also when the percentage of noisy instances increases. So, we are trying an attempt to improve the accuracy of these methods by combing the best parts of methods together and creating a new method with little modification.

Fig. 2: KEEL Tool GUI

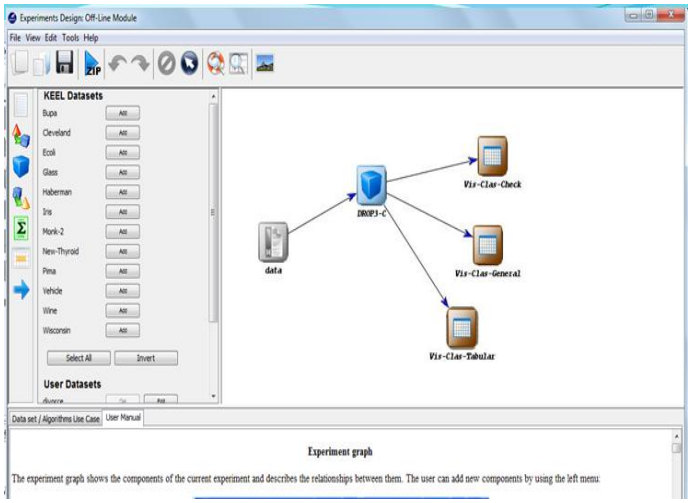

Fig. 3: DROP3 Experiment

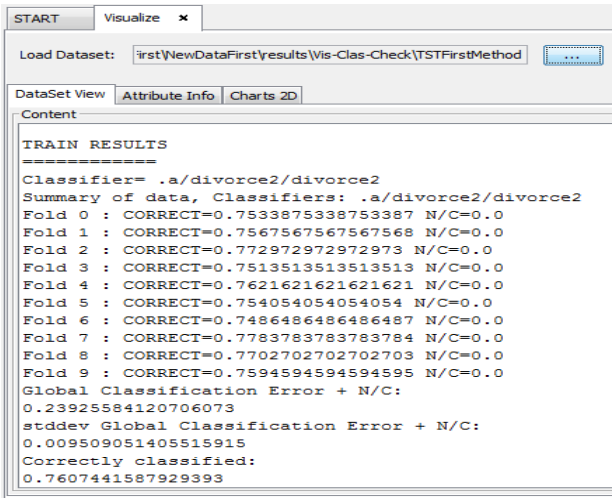

Fig. 4: DROP3 Result

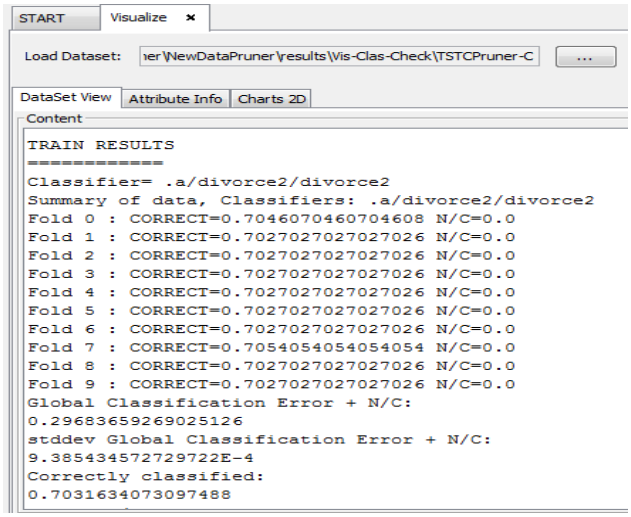

Fig. 5: CPruner Result

IV. PROPOSED APPROACH

The combined approach will be as follows, as the Drop algorithm is weak in removing noisy instances but good at classifying the data properly. And CPruner is good at removing the noisy instances but less efficient at classifying the dataset. We are combing both the algorithms by considering the good qualities of each and replacing their weak parts. The procedure will be as follows:

***Step 1*:** Applying CPruner algorithm to remove the noisy instances. The two rules are:

Rule1-Instance pruning rule

For an instance xi in TR, if it can be pruned, it must satisfy one of the following two conditions:

-It is a noisy instance;

-It is a superfluous instance, but not a critical one.

Rule 2: Rule for deciding the order of instances removal

Let H-kNN(xi) be the number of the instances of its class in kNN(xi), and D-NE(xi) be the distance of xi to

its nearest enemy.

For two prunable instances xi and xj in TR,

If H-kNN(xi) > H-kNN(xj), xi should be removed before xj ;

If H-kNN(xi) = H-kNN(xj) and D-NE(xi) > D-NE(xj), xj should be removed before xi;

If H-kNN(xi) = H-kNN(xj) and D-NE(xi) = D-NE(xj), the order of removal is random decided.

***Step 2*:** Now applying DROP algorithm for classifying the dataset obtained from Step1.

It uses the following basic rule to decide if it safe to remove an instance from the instance set S(where S = TR originally):

Remove xi if at least as many of its associates in S would be classified correctly without xi.

V. FUTURE WORK

This is just an attempt to improve the efficiency of classification of the dataset with accuracy more than 0.76

that was of Drop algorithm. This approach is yet to be tested on our given dataset to compare the results from previous methods. The work is still going to develop the pseudo-code for above algorithm and then registering the new method into KEEL tool to make it available for further work on classification.

VI. CONCLUSION

From above experimental results we have come to know that the two previously developed methods of prototype selection are not much efficient for classification. So we are trying to build a new method by combining two approaches. It will not only remove the noisy instances but also will classify the dataset more accurately. That is it will be able to overcome all the 3 drawbacks of k-NN classifier simultaneously viz. Improving noise tolerance by CPruner step, improving classification efficiency by Drop algorithm step and if noisy instances are reduced and classification is accurate then it will indirectly reduce the storage requirement and time complexity of data mining tasks.

REFERENCES

[1] http://www.inf.ed.ac.uk/teaching/courses/dme/html/datasets0405.html for dataset used in the experiment.

[2] Shikha V. Gadodiya, Manoj B. Chandak, 2013 Prototype Selection Algorithms for kNN Classifier: A Survey in International Journal of Advanced Research in Computer and Communication Engineering Vol.2, Issue 12, December 2013

[3] Garc_IAET AL 2012 , Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 3, March 2012.

[4] Salvador Garc´ıa, Joaqu´ın Derrac, Jos´e Ram´on Cano, and Francisco Herrera, .PROTOTYPE SELECTION FOR NEAREST NEIGHBOR CLASSIFICATION: SURVEY OF METHODS.

[5] K. Hattori and M. Takahashi 2000, A new edited k-nearest neighbor rule in the pattern classification problem, Pattern Recognition, vol. 33,no. 3, pp. 521–528, 2000.

[6] J. ALCALÁ-FDEZ, A. FERNÁNDEZ, J. LUENGO, J. DERRAC,S. GARCÍA, L. SÁNCHEZ AND F. HERRERA , KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework in J. of Mult.-Valued Logic & Soft Computing, Vol. 17, pp. 255–287.

[7] http://sci2s.ugr.es/pgtax/experimentation.php#summ1m to get KEEL software tool.

[8] N. Jankowski and M. Grochowski 2004, Comparison of Instances Selection Algorithms I. Algorithms Survey,Proc. Int'l Conf. Artificial Intelligence and Soft Computing, pp. 598-603, 2004.