Optimization of the Ho-Kashyap Classification Algorithm Using Appropriate Learning Samples

Dezfoulian,Mir Hossein*[1] MiriNezhad,Younes[2] Mousavi,Seyed Muhammad Hossein[3] Shafaei Mosleh,Mehrdad[4] Shalchi,Muhammad Mehdi[5]
Department of Computer Engineering
Bu Ali Sina University
Hamadan, Iran
Dezfoulian@basu.ac.ir , y.mirinezhad93@basu.ac.ir , h.mosavi93@basu.ac.ir

Abstract—**This article is focusing on optimization of the Ho-Kashyap classification algorithm. Choosing a proper learning sample plays a significant role in runtime and accuracy of the supervised classification algorithms, specially the Ho-Kashyap classification algorithm. This article with combining the methods of Multi Class Instance Selection and Ho-Kashyap, not has only reduced the starting time of algorithm, but has improved the accuracy of this algorithm, using proper parameters. The results of this suggested method, in terms of accuracy and time, are evaluated and simulations have proved that MCIS method can choose the data that have more effectiveness on classification, using proper measures. If Ho-Kashyap algorithm classifies using more important data, it could be to save the time in classification process and even increases the accuracy of classification.**

Keywords-component; Classification; Ho-Kashyap; MCIS

I. INTRODUCTION

By the recent technology advancement, specially in the field of computer systems, calculating works are done easier. The classification of data in researches and sciences, in different fields such: medicine, agriculture, marine industry, chemistry, etc, is of high significance. The aid of different classification algorithms, could classify raw data. There is a variety of databases on this purpose and daily efforts to improve the classification methods and propose the new classification algorithms, are done. So far many algorithms for classification have been presented in which, each has had its cons and pros. One of the worse disadvantages of the classification methods is the influence of the database which are far-from-center, or noisy data, on the classifier. There have been many methods proposed to remove the data which have destructive influence on the accuracy of classifiers. One of these methods is Multi Class Instance Selection (MCIS)[1], that has been used in this paper to improve Ho-Kashyap[2].

II. RELATED PREVIEWS WORKS

So far there has been many methods proposed to classify data which each has its cons and pros. In 1963, Support Vector Machine (SVM) algorithm was invented for the first time and in 1995 was generalized by Vapnik and Cortes for the non-linear state[3]. The basis of this algorithm is linear classifying and tries to choose the line with more safety margin. Learning this algorithm is almost easy and works well with high dimensional data. In 1965 Kashyap, R.L and Ho, Y.C. proposed a convergent and limited algorithm for resolving linear inequalities which was useful for solving classification sample problems. They named the proposed algorithm, Ho-Kashyap[2]. This algorithm is so fast but is sensitive to noisy data and far of center data. In 1967 Cover and Hart proposed an algorithm (Knn) which do the classification and approximate to the nearest K data[4]. In 2003 Jacek Leski designed a linear classifier based on approximation error and Ho-Kashyap classification algorithm[5]. This classifier designs the most separatable margins,just like SVM . Of the most disadvantages of this algorithm, its high sensitivity to noise could be mentioned. In 2006 Fabien Lauer and his teammates used the early stop based on generalization of estimating error for preventing Over-Fitting in Ho-Kashyap algorithm[6].

In 2013 Jingnian Chen and his teammates proposed a method that with the use of deleting non-boundary data, improved the SVM[7],[1]. This algorithm is based on check the data distances of each other, and succeeded to identify boundary data.

Prof. Dr. Chris Cornelis and Dr. Yvan Saeys used MCIS method for their investigation called “Fuzzy Rough and Evolutionary Approaches to Instance Selection” to select best instances[8].

Hyunchl Ahn and William X. S. Wong used this method in 2016 for Corporate Credit Rating by Multi Support Vector Mchines with Simultaneous Multi-Factors approach[9].

In April 2016 , B.Ramesh and J.G.RSathiaseelan implement ECAS stemmer, Efficient Instance Selection and Pre-computed Kernel Support Vector Machine for text classification[10] using MCIS method.

These two also used it for Instant Selection for Micro Array Data sets using SVM[11].

III. PROPOSED METHOD

As mentioned before, one of the problems for finding classifier, is the existence of far-from-center data, or noisy data. This paper has used MCIS to improve the performance of Ho-Kashyap and remove the far-from-center data. This means that firstly the existing data are checked by MCIS and the data that are proper to be classified by Ho-Kashyap are chosen. Then classifier is made by decreased data. In addition to the influence of noisy and far-from-center data, on the accuracy of classification, as the number of data decreases, the speed of finding classifiers increases. While there is more data, this speed increasing, is much sensible.

A. MCIS(Multi Class Instance Selection)

First, the data of one class (choosing data is accidental) is considered as positive class, and other data (the data of other classes) are chosen as negative class. Then the positive class's data is clustered by K-Means and the center of every cluster is considered as the representative of that cluster. Then calculate the distance of all negative class's data of positive class's center and those distances that are less than r, are marked as boundary data of the chosen class. It must be regarded, the less r is, the fewer number of the points which are marked as boundary points in a class would be. We do all these actions for all existing classes till when all the boundary data are marked.

B. Classification

After choosing appropriate data using MCIS algorithm, these data are classified using Ho-Kashyap method. Due to this fact that some of data do not attend in classification, it is expected that the time of execution of classification process is less that the time in which Ho-Kashyap algorithm does the classification process using all data and as regards Ho-Kashyap being sensitive to the noisy and far-from-center data, and these data are removed by MCIS, it is expected that accuracy also increases.

IV. TESTS AND RESULTS

To illustrate the influence of choosing proper learning samples on Ho-Kashyap, First, boundary data are identified by MCIS method, then these data are classified by Ho-Kashyap algorithm and compared with classification algorithm of KNN, SVM, MLP, FSM and Ho-kashyap.

A. Datasets

The tests are done using these databases: Pima diabetes, Ionosphere, Haberman, Breast cancer Wisconsin, and in the Table I, number of features, classes and chosen sample for this learning and test is mentioned. In every database, 70 percent of samples for learning and 30 percent for test have been used.

B. Hardware

Due to dependence of hardware to comparison of the runtime of Ho-Kashyap with all data and with selective data, all tests have been done using a system with following features: Intel® Core™ i7-2670QM and 4GB of RAM. The results of these classification algorithms: FSM[12], MLP[13], SVM, KNN, due to the aim of this paper, which is comparing the accuracy of algorithms with the accuracy of Ho-Kashyap algorithm, have been extracted from related articles and websites.

Table I . Datasets and features

	Pima diabetes	**Ionosphere**	**Haberman**	**Breast cancer Wisconsin**
Samples	768	351	306	699
Features	8	33	3	01
Choosing Features	8	33	3	01
Classes	2	2	2	2
Learning Samples	538	246	214	384
Test Samples	231	011	42	201

C. Experimental results

This research firstly studied the Ho-Kashyap algorithm in order to indicate the influence of removing far-from-center data. As it is illustrated in the Table II, after removing far-from-center data (data with distances more that r), the accuracy of algorithm increases and the runtime decreases. The result of this would be illustrated better using a database with higher dimensions and with specifying proper r and k the accuracy could be increased besides decreasing runtime.

Table II . Comparison of Ho-Kashyap algorithm with all datasets and Ho-Kashyap with selected instances

	Pima diabetes k = 10,r = 4	**Ionosphere K = 2 and r = 9**	**Haberman K = 4 and r = 1**	**Breast cancer Wisconsin K = 3 r =3.5**
Accuracy of Ho-Kashyap	76.45%	87.2%	73.8%	96.2%
Runtime of Ho-Kashyap	0.76 second	1.91 second	0.2 second	0.76 second
Accuracy of MCIS + Ho-Kashyap	78.47%	88.9%	75.7%	97.38%
Runtime of MCIS + Ho-Kashyap	0.62 second	1.54 second	0.15 second	0.58 second

For the comparison of the Ho-Kashyap accuracy with other classification algorithms, after removing the far-from-center data, in Table III, the result of every algorithm on every datasets which are extracted from www.is.umk.pl is shown.

Table III . Comparison of suggested method with other classification algorithms

	MCIS + Ho-Kashyap	Ho-Kashyap	SVM	KNN	MLP	FSM
Pima diabetes	78.47%	76.45%	77.5%	76.7%	76.4%	75.4%
Ionosphere	88.9%	87.2%	93.0%	98.7%	96.0%	92.8%
Breast cancer Wisconsin	97.37%	96.2%	96.9%	97.1%	96.7%	98.3%

V. CONCLUSION

With respect to information of tables, this could be deduced that the more there are far-from-center data in a database, the less the accuracy of classification is, and by deleting those data, the accuracy of classification improves. Moreover, the more number of data and their dimensions is, the runtime of classification algorithm will increase. Therefore, when some of data are put away as far-from-center data, the process of classification will be done with a fewer number and the speed of classification will increase. Of course, the increase of classification speed will be more obvious as using a big data.

REFERENCES

[1] Chen, Jingnian, et al. "Fast instance selection for speeding up support vector machines." *Knowledge-Based Systems* 45 (2013): 1-7.

[2] Ho, Yu-Chi, and R. L. Kashyap. "An algorithm for linear inequalities and its applications." IEEE Transactions on Electronic Computers 5.EC-14 (1965): 683-688.

[3] Cortes, Corinna, and Vladimir Vapnik. "Support vector machine." Machine learning 20.3 (1995): 273-297.

[4] Cover, Thomas M., and Peter E. Hart. "Nearest neighbor pattern classification." Information Theory, IEEE Transactions on 13.1 (1967): 21-27.

[5] Łęski, Jacek. "Ho–Kashyap classifier with generalization control." Pattern Recognition Letters 24.14 (2003): 2281-2290.

[6] Lauer, Fabien, and Gérard Bloch. "Ho–Kashyap classifier with early stopping for regularization." *Pattern recognition letters* 27.9 (2006): 1037-1044.

[7] Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition." Data mining and knowledge discovery 2.2 (1998): 121-167.

[8] Verbiest, Nele. *Fuzzy rough and evolutionary approaches to instance selection*. Diss. Ghent University, 2014.

[9] Ahn, Hyunchul, and William XS Wong. "Multiclass Support Vector Machines with Simultaneous Multi-Factors Optimization for Corporate Credit Ratings."*World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering* 10.4 (2016): 643-647.

[10] Ramesh, B., and J. G. R. Sathiaseelan. "AN IMPLEMENTATION OF EIS-SVM CLASSIFIER USING RESEARCH ARTICLES FOR TEXT CLASSIFICATION." *ICTACT Journal on Soft Computing* 6.3 (2016).

[11] Ramesh, B., and J. G. R. Sathiaseelan. "Support vector machine using efficient instant selection for micro array data sets." *Computational Intelligence and Computing Research (ICCIC), 2014 IEEE International Conference on*. IEEE, 2014.

[12] Zeng, Zheng, Rodney M. Goodman, and Padhraic Smyth. "Learning finite state machines with self-clustering recurrent networks." Neural Computation5.6 (1993): 976-990.

[13] Pal, Sankar K., and Sushmita Mitra. "Multilayer perceptron, fuzzy sets, and classification." Neural Networks, IEEE Transactions on 3.5 (1992): 683-697.