VOL.3 NO.2September 2014 - ISSN 2233 – 1859

UOIuBIH
ORSinBIH
Operations Research Society in Bosnia and Herzegovina

Southeast Europe Journal of Soft Computing

Available online: www.scjournal.com.ba

IUS Soft Computing Research Group

Chromosome Polarity Determination Based on the Total Length and Centromere Location Using Machine Learning Algorithms

Kanita Karaduzovic-Hadziabdic and Sadina Gagula-Palalic
International University of Sarajevo, Faculty of Engineering and Natural Sciences, Hrasnicka Cesta 15, Ilidža 71210 Sarajevo, Bosnia and Herzegovina

Article Info

Article history:
Received Sep.2014
Received in revised form Nov 2014

Keywords:
Randomized algorithms, sorting problems, linear time algorithm

Abstract

In this work we determine chromosome polarity based on three machine learning methods: multilayer perceptron (MLP) neural networks, k-nearest neighbor (*k-nn*) method and support vector machines (SVM). In all three machine learning methods only two chromosome features, total length of the chromosome and the cetromere location, were used to determine the chromosome polarity. Classification results obtained are 95.94%, 95.255%, and 95.88% for MLP neural networks, k-nn method and SVM respectively.

1. INTRODUCTION

Chromosome analysis is widely used in cancer diagnosis, genetic disorders, as well as in genetic research in laboratories. Various automated studies have been performed for chromosome analysis (Ledley and Ruddle 1996; Guimaraes et al., 2003). Many of computer-aided methods for chromosome analysis are based on machine learning approaches (Wu et al. 1990; Sweeney et al. 1993; Ruan 2000; Cho 2000; Can et al. 2012, Karaduzovic-Hadziabdic 2012). A human karyotype contains 46 chromosomes belonging to 22 pairs of classes and two sex chromosomes. One of the main features characterizing a chromosome is its length, which needs to be normalized since it varies depending on the phase of the cell division. The best time to measure the chromosome length is the metaphase. The chromosome length used in this work has manually been computed by an expert. Each chromosome contains a p-arm (shorter, top arm), and a q-arm (longer, lower arm). Based on the size of these two arms chromosomes may be metacentric (when two arms are almost equal), submetacentric or acrocentric (when two arms are not equal, and the shorter arm is referred to as the p-arm). Centromeric index is another characteristic chromosome feature. It is the ratio of the p-arm length to the whole length of the chromosome. In this work the data for the p-arm length were obtained manually by an expert. The task of polarity determination is to determine the chromosome orientation, where the standard is that the p-arm is the ‘upper’ arm. An expert decides on polarity based on the lengths of arms and band pattern profiles. In this work, this task has been performed by applying three machine learning methods that use two chromosome features for determining polarity: chromosome length and centromere location (i.e. assumed p-arm length). The initial data set consists of two input features (p-arm length and total length), and one output (polarity), which is equal to 1. Randomly, this set was divided into two parts, where first part remained the same, while the second part of input-output data was processed such that, p arm length was replaced by q-arm length, and output polarity was changed to -1. In this way, we artificially created samples of negative polarity, so that the training data may contain both examples. Finally, the training and testing was performed using three different algorithms to check if those could be used for chromosome polarity prediction.

Not much research has been performed on chromosome polarity determination. Wang et al. (2007) achieved the cell based polarity classification and they achieved an accuracy of 97.4% using their own chromosome data set. Piper et al., (1989) obtained chromosome polarity classification of 96.0%, 94.4%, and 90.6% using the Copenhagen, Edinburgh, and Philadelphia dataset respectively. In our work we perform and compare the classification results using three machine learning methods, MLP neural networks, k-nn method and SVM method using the Sarajevo Chromosome Data Set (Gagula-Palalic, 2013). Out of the three methods, MLP approach achieves the best results, 95.94%.

Figure 1. illustrates the class distribution of normal and reversed chromosome polarities. The x axis corresponds to the p-arm length and the y axis corresponds to the total length of the chromosomes.

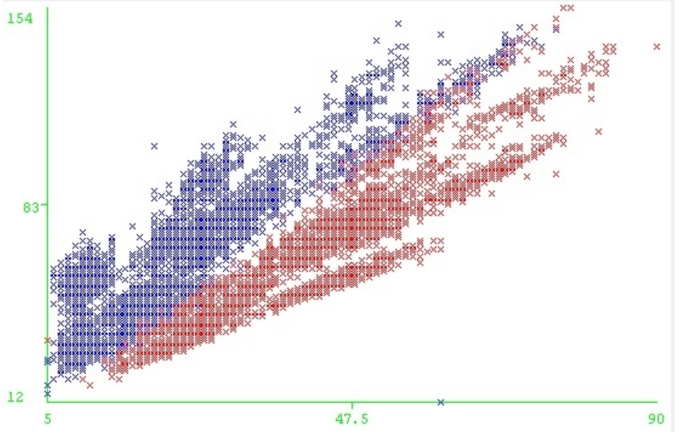

Figure1 Class distribution of positive polarity orientation represented by blue points on the graph, and negative polarity orientation represented by the red points.

2. DATABASE FEATURES

Data used throughout the work was obtained from the Sarajevo Chromosome Data Set (Gagula-Palalic, 2013). Images of the metaphase chromosomes were obtained at the Clinical Center of University of Sarajevo. Images were processed, chromosomes were segmented and several features (total length, p-arm length and band pattern profiles) are extracted by using methods as proposed in (Gagula-Palalic, and Can, 2012a, and 2012b). In this work, we propose to use only two features, which are total chromosome length and p-arm length. A total of 8497 data samples were used in the experiments, containing all 24 chromosome classes. Two input features were supplemented by an output which is a correct chromosome polarity. Some data processing was then performed to artificially obtain the reversed chromosome polarities for half of the data, taking into consideration to split the samples equally for each chromosome type. Therefore, the data was either in form:

(total_length, p_arm_length, +1) for positive polarity, OR
(total_length, q_armlength, -1), for negative polarity

3. METHODOLOGY

3.1 Multilayer perceptron (MLP) neural networks

This section provides a brief overview of neural networks as one of the methods used in classification of chromosome polarity. Neural network consists of artificial nodes (neurons) that are interconnected forming a network. Initially, neural network is trained by the learning process. In supervised learning, the weights between the connected neurons are adjusted such that the difference between the desired response and the actual response are minimized. Once the convergence is obtained (i.e. there is no significant improvement in the difference between the desired response and the actual response) the knowledge stored in the weights will be used during the test phase to perform data classification (Haykin 1999). MLP neural network trained with a back-propagation algorithm is one of the most widely used classifiers reported in literature. The algorithm consists of two passes, forward pass and a backward pass. During the forward pass the error between the desired and the actual response is calculated and is used to update the weights in the backward pass.

MLP neural network consists of the input layer, one or more hidden layers and an output layer. The number of input and output neurons in the input and output layers respectively are defined in the task specification. The number of hidden layers is usually task dependant. By experiment, it was found that two hidden layers produce best results for chromosome polarity classification. Figure 2. Illustrates the architecture of the MLP neural network. The inputs to the neural network correspond to the two characteristic chromosome features (total length of the chromosome and the centromere location).

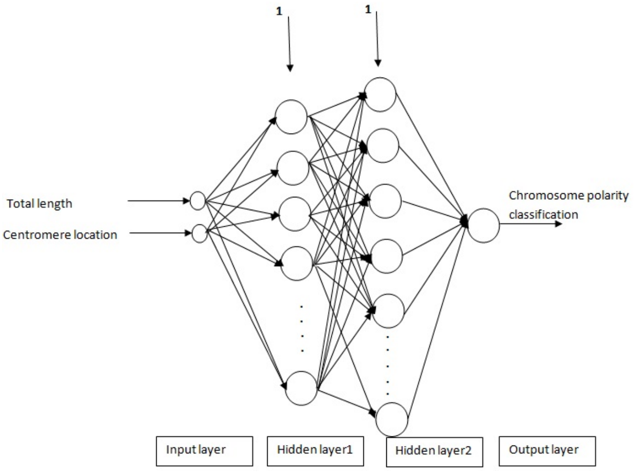

Figure 2 Architecture of MLP neural network

TABLE 1. OVERVIEW OF MLP ARCHITECTURE PARAMETERS

Network type	Feed forward MLP trained by back-propagation algorithm
Learning rate	0.3
Momentum	0.2
Number of input neurons	2, the mentioned features
Number of hidden layers	2
Activation function	non-linear anti-symmetric sigmoid hyperbolic tangent function

The classification accuracy obtained using the MLP neural network with the parameters as outlined in Table 1 was 95.94%.

3.2 k-nearest neighbour (k-nn) method

Chromosome polarity classification was also performed using the *k-nn* classifier. By using this method of classification, a data sample is classified depending on the majority vote of its nearest k training samples in the feature space. By using a predefined distance function, distance of a data sample to its neighbors is determined. Various distance functions can be used in *k-nn* classifier (Karaduzovic-Hadziabdic, 2012). In our experiments we applied L_p norm distance function. When p is set to 1, Manhattan distance (the L_1 norm) is obtained and when p is set to 2, Euclidian distance function is obtained.

L_p norm distance function is defined as:

$$L_p(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{1/p} \qquad (1)$$

When p = 2, we get the Euclidean distance (L_2 norm):

$$L_2\,(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^2\right)^{1/2} \qquad (2)$$

When p = 1, we get the Manhattan distance, (L_1 norm):

$$L_1(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|\right) \qquad (3)$$

We performed several experiments to test which distance function and which k value achieves the best results for our classification purpose. The optimal classification result of 95.255% was obtained when k was set to 3 and Manhattan distance function was used.

3.2 Support Vector Machines (SVM)

Support vector machines are types of machine learning algorithms which perform both linear and nonlinear classification and regression. SVM utilize different kind of kernels for an efficient nonlinear classification and has been shown empirically to be very successful tool for classification. SVM works in such a way that it separates classes by a hyperplane by maximizing the margin and minimizing the classification error (Haykin, 1999). In this work, two different kernels were utilized for creating SVMs, Polynomial kernel (Equation 3.1, with $p = 1$) and RBF kernel (Equation 3.2, with $gamma = 0.01$):

$$K(x,y) = \langle x,y\rangle^p \qquad (4)$$

$$K(x,y) = e^{-gamma*\langle x-y,x-y\rangle^2} \qquad (5)$$

4. RESULTS AND DISCUSSION

This section presents and discusses the performance of three machine learning methods in determination of chromosome polarization. Table 2 summarizes the results of the performed classification experiments using the methods mentioned is Section 3. The table shows the achieved classification accuracy, root mean square error and ROC area. From the obtained results, it can be seen that the SVM method based on the PolyKernel kernel achieves the best results, 95.88%, from the three methods tested.

TABLE 2. CLASSIFICATION RESULTS

Method	**Accuracy**	**RMSE**	**ROC Area**
MLP neural network	95.94%	0.167	0.995
k-nn (where k=3 with Manhattan distance function)	95.255%	0.179	0.99
SVM (PolyKernel) p=1	95.88%	0.203	0.959
SVM (PolyKernel) P=1.5	95.502%	0.212	0.955
SVM (PolyKernel) P=2	92.629%	0.272	0.926
SVM (RBF Kernel)	85.84%	0.376	0.858

Table 3. displays the results achieved by the *k-nn* method when different values of k and two distance functions (Euclidian and Manhattan distance functions) are used.

TABLE 3. K-NN METHOD RESULTS

k	**Euclidian distance function (Accuracy %, RMSE, ROC)**	**Manhattan distance function (Accuracy %, RMSE, ROC)**
1	95.0312, 0.195, 0.986	95.055, 0.195, 0.986
2	95.196, 0.184, 0.989	95.219, 0.182, 0.989
3	95.243, 0.118, 0.99	**95.255, 0.179, 0.99**
4	95.125, 0.178, 0.991	95.125, 0.178, 0.991
5	95.137, 0.177, 0.992	95.137, 0.177, 0.992

Figure 2 graphically illustrates Table 3 results. The graph represents the accuracy comparison between Euclidian and Manhattan distance functions for different values of k. Increasing the value of k does not result in the improved performance for k values greater than 3. Furthermore, Euclidian distance functions and Manhattan distance functions yield the same results for k values greater than 3.

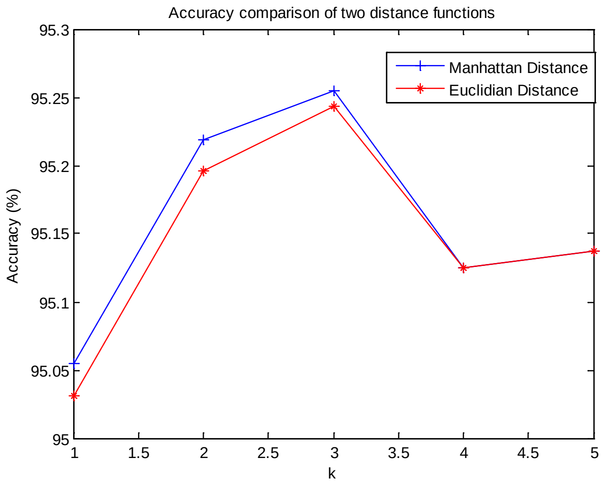

Figure 3. Accuracy comparison between Euclidian and Manhattan distance functions for different values of *k*

Table 4. Confusion matrix for MLP neural network

4044 True Positives	204 False Positives
141 False Negatives	4104 True Negatives

Table 5. Confusion matrix for SVM using PolyKernel kernel

4061 True Positives	187 False Positives
163 False Negatives	4082 True Negatives

Table 6. Confusion matrix for k-nn method (k=3, using Manhattan distance function)

4083 True Positives	165 False Positives
238 False Negatives	4007 True Negatives

Tables 4-6 represent the confusion matrix for each of the tested machine learning methods of chromosome polarity classification. Confusion matrix contains the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values of the performed classification. These values can be calculated using the following formulas:

5. CONCLUSION

In this work we described and evaluated the results of the classification of chromosome polarity orientation using three machine learning methods, MLP neural networks, k-nn method, and SVM. It was found that the best classification performance is achieved using the SVM method, 95.88%. All of the experiments were performed using the Sarajevo Chromosome Data Set using only two chromosome features (total chromosome length and centromere location). Furthermore, for the k-nn method, we analyzed the performance of two widely used k-nn distance functions, Euclidian and Manhattan distance functions and found that Manhattan distance functions with *k* set to 3 achieves the best results.

In the future work, we may include features derived from the band pattern profiles of chromosomes and test the performance of polarity determination using the above algorithms.

6. REFERENCES

Can, M., and S. Gagula-Palalic (2012), Application of Ensemble Machines of Neural Networks to Chromosome Classification, Southeast Journal of Soft Computing, Vol.1, No.2, pp. 31-35.

Cho J.M. (2000) Chromosome Classification using backpropagation neural networks, IEEE Engineering Medicine and Biology, January/February 2000, pp. 28-34

Gagula-Palalic, S. (2013), Database: http://www.ius.edu.ba/Default.aspx?PageContentID=1817&tabid=192

Gagula-Palalic, S., & Can, M. (2012). Automatic segmentation of human chromosomes. *Southeast Journal of Soft Computing*, 1(2), 80–83.

Gagula-Palalic, S., & Can, M. (2012). Extracting Gray Level Profiles of Human Chromosomes by Curve Fitting. *Southeast Journal of Soft Computing*, 1(2), 66–71.

Guimaraes, L. V., Schuck, A., Elbern, A., (2003) Chromosome classification for karyotype composing applying shape representation on wavelet packet transform, Engineering in Medicine and Biology Society, 2003, Proceedings of the 25th Annual International Conference of the IEEE, vol. 1, pp. 941 – 943.

Haykin S. (1999) Neural networks and Learning Machines, Pearson, 2nd Ed.

Ledley R S, and Ruddle F H (1966) Chromosome analysis by computer, Scientific American, Vol. 214, No 4, pp 40-46.

Lindenbaum L, Markovitch S, Rusakov D (2004), Selective sampling for nearest neighbor classifiers, Machine Learning, pp. 125–152.

Karađuzovć – Hadžiabdić K., (2012) “Classification of chromosomes using nearest neighbor classifier”, *Southeast Journal of Soft Computing*, Vol 1, No2, pp.12-15.

Ruan X (2000), “A Classifier with the Fuzzy Hopfield Network for Human Chromosomes”, Proceedings of the 3rd World Congress on Intelligent Control and Automation, June 28 - July2, pp. 1159 - 1164.

Sweeney Jr W P, Musavi M T, Guidi J N (1993), “Probabilistic Neural Network as Chromosome Classifier”, Proceedings of 1993International Joint Conference on Neural Networks, Vol 1, pp. 935-938

Wu Q, Suetens P, Oosterlinck A (1990), “Chromosome classification using a multi-layer perceptron neural net ”, Annual International Conference of the IEEEE Engineering in Medicine and Biology Society, 1990, Vol.12, No. 3, pp. 1453-1454.