SOUTHEAST EUROPE JOURNAL OF SOFT COMPUTING
Available online at www.scjournal.com.ba

Classification of chromosomes using nearest neighbor classifier

Kanita Karaduzovic-Hadziabdic

Faculty of Engineering and Natural Sciences, International University of Sarajevo,
Hrasnicka Cesta 15, Ilidza, 71200 Sarajevo BiH, kanita@ius.edu.ba

***Abstract*— This paper addresses automated classification of human chromosomes using *k* nearest neighbor classifier. *k* nearest neighbor classifier classifies objects according to the closest training sample in the feature space. Various distance functions can be used in computation of how close the object is to the training sample. In this work various different distance functions are used to compare the performance of each. It was found that Euclidean distance function produces the best results.**

Keywords— nearest neighbor classifier, chromosome classification

INTRODUCTION

Chromosome classification can be used in pre-natal diagnosis of genetic disorder, some cancer diagnosis or bone marrow transplant studies. A human cell contains 46 chromosomes belonging to 24 classes. These 46 chromosomes consist of 22 pairs of classes, and two sex chromosomes. A traditional method of human chromosome classification is done by keryotyping, classification by inspection under a microscope by a human expert. This method of classification takes about 10 minutes, with an error of 0.3% (Ritter and Gallegos, 1997).

One of the difficulties in chromosome classification is the chromosome variability within one chromosome class, originating from different metaphases. Another difficulty is that chromosomes may overlap and touch each other, may be bent, and have different orientation. Furthermore, a high number of classes that need to be differentiated adds to the complexity of the task.

Since mid-sixties automated analysis was initiated by Ledley and Ruddle (1966). Some of these computer-aided techniques include parametric classifiers (Oosterlinck et al. 1997), maximum likelihood classifiers (Piper 1987), and Markov networks (Granum et al. 1989). Achieved accuracy using these methods was only between 75% - 85%. Machine learning methods have been widely employed in classification tasks. One of their main advantages is efficiency in dealing with a large amount of data. As chromosome classification is a pattern recognition problem, different artificial neural network methods have been employed in their classification: multi-layer perceptron [Wu et al. 1990; Delshadpour 2003], probabilisitc neural networks [8], neuro-fuzzy classifier (Ruan 2000), etc.

In this paper *k*-nearest neighbor (*k*-nn) method is used in chromosome classification and a comparative study of different distance functions is performed. The steps taken during this research are as follows:

- Data retrieval
- Feature selection
- Division of the data into training and test sets
- Applying the *k*-nn classifier using
    - Euclidean distance
    - P-norm
    - Mahalanobis distance

DATABASE AND FEATURES

Data used throughout this study was obtained from the Copenhagen database [Lundsteen et al. 1980; Granum and Thomason 1990], where chromosome density profiles were extracted from images of cells in the metaphases stage of cell division. The success rate of classification highly depends on the quality of the dataset. The quality of Copenhagen dataset chromosomes is considered to be good since the chromosomes were measured carefully using densiometry of photographic negatives from selected high quality cells. All the classifications of the chromosomes in the Copenhagen dataset were classified by a cytegeneticist. None of the chromosomes from this dataset exhibit any abnormalities.

The Copenhagen dataset was pre-processed where all the text features were converted to digits for further processing. A total of 4400 data samples were used in the experiments carried out in this research, and the data was divided into two parts. 2200 samples were used for the training purposes (100 data samples for each chromosome class) and a remaining 2200 for testing purposes.

The features used include the chromosome length, centromere index and the gray banding pattern. The longest chromosome in the dataset used consisted of 100 bands in the banding profile, thus the feature space for each chromosome consists of 102 numbers.

K NEAREST NEIGHBOR CLASSIFIER

Chromosome classification was carried out by *k* nearest neighbor (*k*-nn) classifier. *K*-nn is one of the most popular classification method mainly due to its ease of implementation and succussful classification results. A sample is classified according to the majority vote of its nearest *k* training samples in the feature space. Distance of a sample to its neighbors is defined using a distance function.

For all points x, y, and z, a distance function F(., .), must satisfy the following:

- nonnegativity: $F(x, y) \geq 0$
- reflexivity: $F(x, y) = 0 \; if \; and \; only \; if \; x = y$
- symmetry: $F(x, y) = F(y, x)$
- triangle inequality $F(x, y) + F(y, z) \geq F(x, z)$

Three distance functions that can be used in *k*-nn classifier are:

L_p norm:

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

- Euclidean distance, L_2 norm:

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

- Manhattan or city block distance, the L_1 norm:

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

- Mahalanobis distance that takes into account the correlation *S* of the dataset :

$$L_m(x, y) = \sqrt{(x - y)S^{-1}(x - y)} \quad (4)$$

In the experiments carried out in this research *k* was taken to be 1.

The steps that need to be carried out during the *k*-nn algorithm are as follows:

- divide the data into training and test data
- select a value *k*
- determine which distance function is to be used
- choose a sample from the test data that needs to be classified and compute the distance to its *n* training samples
- sort the distances obtained and take the *k* nearest data samples
- assign the test class to the class based on the majority vote of its *k* neighbors

Despite its ease of use the two main drawbacks of the nearest neighbor classifier are:

1. High computation cost: during computation, a distance between the test sample and *all* of the stored training samples must be calculated one by one and a list of the *k* closest ones is kept. Reducing the training set reduces the rate of successful classification, however, increasing the training set increases the computation time. One approach to overcome this problem is to reduce the dimensions of the feature space by using principal component analysis. Another approach is to modify the training set by removing some samples that belong to the same class label and exibit similar features.
2. The algorithm performance depends on the training set used. If the training data set is not representative enough then poor classification results may be obtained.

[Anil 2006; Lindenbaum et al. 2004] describe some techniques that try to overcome these problems.

RESULTS

Table 1. Results of classification experiments using various p-norm distance functions

Distance Function	Accuracy
p-norm, p = 0.5	93.73%
p-norm, p = 1	94.73%
p-norm, p = 2 (Euclidean distance)	95.18%
p-norm, p = 3	93.41%
p-norm, p = 6	88.36%
p-norm, p = 20	86.59%

The above table shows a gradual rise in the success rate starting from p-norm 0.5 until p-norm 2 (i.e. the Euclidean distance) is reached, followed by a study fall.

Experiments were also carried out using Mahalanobis distance function. However, the Mahalanobis distance function is computationally very expensive. The time taken to complete the calculations is considerably longer than the time needed for the p-norm distance functions. To overcome this problem, principal component analysis (PCA) was carried out. Principal component analysis is a common method applied to reduce data dimensions without losing too much information.

Principal component analysis transforms the original data such that the new data has the same number of variables, but most of the variation of the original data is covered by a small number of components. Since the longest gray-level band contains 100 numbers, PCA was performed on those 100 gray-level band values. Only the first 10 principal components that cover most of the data variance were taken into the computations, reducing the number of gray level band numbers taken into computation from 100 to 10. After performing PCA, data input for each chromosome consisted of 12 numbers: 10 numbers for the gray bands, one number for chromosome length and one for centromeric index. This reduction of data dimensionality substantially decreased the performance time. The classification success rate achieved when Mahalanobis distance function was used was 88.01%. Applying this data with reduced dimensionality and using Euclidean distance function resulted in 94.05% success rate, (compared to the previous success rate of 95.18% when no data reduction was applied). Obviously, this difference in the success rate is since after applying PCA, not all of

the available gray bands data of a chromosome was taken into the account.

CONCLUSION

In this research the classification of the chromosomes obtained from the Copenhagen dataset was done using the *k*-nn classifier. The results achieved using Euclidean distance function, p-norm distance function and Mahalanobis distance function were obtained and compared.

Since computing the results by Mahalanobis norm was computationally expensive, principal component analysis was used to reduce the data dimensionality and thus speed up the computation process. In order to compare the results achieved by Mahalanobis distance function (88% classification success rate), where the feature space was significantly reduced, with the ones previously achieved by different p-norms, the classification of chromosomes was also carried out with the best performing distance function, i.e. the Euclidean distance function (94.05% classification success rate), by using the same reduced feature data set as in the Mahalanobis distance. Once again a much better classification results were obtained by using Euclidean distance function.

Since the Euclidean distance function produced the best results, it is not a surprise that it is the most widely used distance function when *k*-nn classifier is used.

ACKNOWLEDGMENT

I would like to thank proof. dr. Mehmet Can for his guidance throughout this research. I would also like to thank my colleague, Sadina Gagula-Palalic for providing the processed Copenhagen chromosome dataset used in this work.

REFERENCES

Anil K G (2006), “On optimum choice of k in nearest neighbor classification”, *Computational Statistics and Data Analysis*, pp. 3113–3123.

Delshadpour S (2003), “Improved MLP neural network as chromosome classifier”, IEEE EMBS Asian-Pacific Conference on Biomedical Engineering, pp 324-325.

Granum E, Thomason M G, and Gregor J **(**1989**)**, ”On the use of automatically inferred Markov network for chromosome analysis.”, Automation of Cytogenetics, Editors C. Lundsteen and J. Piper, Springler-Verlag, Heidelberg, Germany, pp 233-251.

Granum E and Thomason M G (1990), "Automatically inferred Markov network models for classification of chromosomal band pattern structures", Cytometry, Vol. 11, pp 26-39

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.

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.

Lundsteen C, Phillip J, Granum E (1980), "Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes", Clinical Genetics, Vol. 18, pp 355-370

Oosterlinck A, Van Daele J , De Boer J , Dom F, Reynaerts A, Van Den Berghe H (1997), “Computer-assisted karyotyping with human interaction.” The Journal of Histochemistry and Cytochemistry, Vol. 25, No. 7, pp 754-762.

Piper J (1987), “The effect of zero feature correlation assumption on maximum likelihood based classification of chromosomes.”, Signal Processing, pp. 49-57.

Ritter R, Gallegos M T (1997), “Outliers in Statistical Pattern Recognition and an Application to Automatic Chromosome Classification”, Pattern Recognition Letters 18, pp 525-539.

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.

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.