Biohouse, Volume 2: Issue 2, Mar-Apr 2016

Biohouse Journal of Computer Science

ISSN 2379-1500

A Hybrid GA/kNN/SVM Algorithm for Classification of data

Dr. J.Subash Chandra Bose
Assistant Professor
College of Computer Science
Department of Computer Engineering
King Khalid University
Aseer, Saudi Arabia.
jsubashme@gmail.com

Dr. Suresh Babu Changalasetty
Associate Professor
College of Computer Science
Department of Computer Engineering
King Khalid University
Aseer, Saudi Arabia.

Dr. Ahmed Said Badawy
Vice Dean
College of Computer Science,
Department of Computer Engineering
King Khalid University
Aseer, Saudi Arabia.

Dr. Wade Ghribi
Assistant Professor
College of Computer Science
Department of Computer Engineering
King Khalid University
Aseer, Saudi Arabia.

Dr. Jamel Baili
Assistant Professor
College of Computer Science
Department of Computer Engineering
King Khalid University
Aseer, Saudi Arabia.

Mr. Harun Bangali
Lecturer
College of Computer Science
Department of Computer Engineering
King Khalid University
Aseer, Saudi Arabia.

Abstract: *This paper proposes an effective comparison of different algorithms for classification namely Genetic Algorithm, K-Nearest Neighbor (kNN) and Support Vector Machines (SVM) techniques. The goal of the comparison is to compare the effects of the classification rules from data. The algorithm is stimulated by the behavior of classification. Individual data are selected based on how well they fit to the environment. In GP, the entity that reflects this degree of adaptation is the fitness function. The programs that better solve the problem at hand will receive a better fitness value, and will consequently have a better chance of being selected. The datasets that are considered can be any data involving choice of a fitness function, and an evaluation method depends on the problem metrics given to the simulation environment based on parameter values. SVM and KNN classifiers are compared and applied their performance using MATLAB simulation environment.*

Keywords— *K nearest neighboring, Genetic Algorithm, Support Vector Machine, Micro array, Classification.*

1. INTRODUCTION

Nowadays, people can obtain the expression datasets of thousands of genes simultaneously using microarray technology. One of the important fields in using these gene expression datasets is to classify and predict the diagnostic category of a sample. Actually, precise diagnosis and classification is crucial for successful treatment of illness.

For classifying microarray data, one can use the classical liner discriminant analysis, artificial neural networks, KNN, as well as some more sophisticated machine learning methodologies including bagging, boosting and kernel methods. Among them, SVM is one of the most powerful supervised learning algorithms in gene expression analysis. SVM has been found generalization ability and useful in handling classification tasks in case of the high dimensionality and sparsity of data points.

SVM constructs an optimal hyper plane from a small set of samples near the boundary and is sensitive to these boundary samples. The samples intermixed in another class or in the overlapped boundary region may cause the decision boundary too complex and may be harmful to improve the precision of classifier. The existence of samples in the overlapped region may also increase the computation burden and decline the generalization ability of classifier. In addition, labeling a sample in some cases can be subjective and a few mislabeled samples could deeply degrade the performance of the classifier. Mislabeled and troublesome learning samples may be often near the boundary and lead to a result with high error rate. Many researches have been focused on identifying and pruning the questionable redundancy samples to improve the performance of classification.

Though SVM has been found useful in handling classification tasks, it has been recognized that results of SVM analysis can be improved when combining with other classifiers.

A. Overview

Here K-Nearest Neighbor (KNN) classifier and Support Vector Machine (SVM) classifier are applied for one simple reason that the KNN is restrictive whereas SVM does not need a predefined value. If the performance of KNN applied on subset selected using Genetic Algorithm (GA) is better than that of SVM applied on subset selected using Genetic Algorithm (GA) then the performance of SVM has to be further improved. The ultimate goal is to find a better classifier than that are already used in gene data classification.

B. Objective

The objective of the project is to apply classifiers on a subset of gene data selected using a different algorithm. The project aims at providing a classifier with:

(i) High Accuracy
(ii) Low cost
(iii) No optimization problems

In the project first a simple SVM linear classification is made on a gene dataset whose subset is selected through Genetic Algorithm. Similarly KNN is applied on the subset of gene dataset. If the SVM performs better then it is again applied on a larger dataset and compared to more modern classifiers. If the KNN performs better on this not too large dataset then other kernel functions and boosting techniques are applied to SVM to improve its performance.

2. LITERATURE SURVEY

Table 1: An overview of Literature survey

No.	Method	Description
1	Genetic Algorithm[16]	The algorithm is used for particular feature selection (and several association of features) problem encountered in genetic analysis of different diseases.
2	SVM(Support Vector Machine) classifier[12]	The algorithm is well suited for classification and prediction of cancer cells by minimizing loss of data and reduces redundant feature measures.
3	Decision Tree Algorithm[11]	The algorithm maximally diversifies trees in the ensemble decision tree committee. The diversity improves classification accuracy.
4	SVM-RFE(Recursive Feature Elimination)[9]	This is one of the gene selection algorithms which is used to minimize an error bound and improves the performance.
5	Markov Blanket Embedded Genetic Algorithm(MBEGA)[5]	This approach gives better performance than existing counterparts in terms of these criteria: classification accuracy, number of selected genes, computational
6	Genetic Programming classifier[6]	The approach provides better accuracy with majority voting than accuracy of rules in the voting group.
7	Naïve Bayes classifier[7]	The classifier gives better classification accuracy on gene expression data and can also do well with relatively small set of features.
8	KNN classifier[8]	The classifier is used in regulating genes based on the assumption: every gene in the dataset is regulated by at least another gene from the same dataset.
9	Repeat Based Naïve Bayes Classifier[15]	The classifier gives better performance than naïve bayes and does not require additional parameters and performs well on large and skewed datasets.
10	Genetic Algorithm[4]	The GA has proven to be an excellent strategy for selecting the optimal sub-set of genes that makes the perception an optimal classifier.
11	Hybrid classifier[3]	The classifier is used in hand written recognition which provides good recognition rate in training and test datasets.
12	ARC(Automated Resource Classifier)[14]	The classifier is an open source software meeting the user requirements of flexibility.
13	SVM(Support Vector Machine) classifier[10]	The classifier reduces the entity recognition problem by classifying single words.
14	Naïve Bayes classifier[13]	The classifier classifies the housekeeping and tissue specific genes on the basis of physical characteristics alone.
15	SVM(Support Vector Machine) classifier[1]	The classifier modifies the traditional gauss kernel by adding a positive constant to the kernel function and take weighted SVM when training datasets of the Two-way scenario is imbalanced.
16	BSTC(Boolean Structure Table Classifier)[2]	BSTC extends generalized CAR-based methods to larger datasets. It easily generalizes to multi-class gene expression datasets.

3. PROPOSED METHODOLOGY

In this project, KNN and SVM are proposed to deal with the problem of classification of datasets and hence compare the classification accuracy of the two algorithms to check their performance. KNN is a very efficient pattern recognition method and can be easily carried out. In a statistical opinion, the error rate of a KNN classifier tends to the Bayes optimal when k and the size of sample set tend to infinity. Based on these advantages, KNN is introduced to classify a gene expression dataset. Firstly Genetic algorithm (GA) is used to select the best features and train the obtained features with KNN to prune training samples and then separately classify the best features with SVM to improve the classification. The proposed algorithm

ISSN 2379-1500

can be used in binary and multi-class classification of gene expression data. The results were compared to those obtained by single SVM and KNN. It has been demonstrated that the proposed method is a useful tool for classification and the classification performance is stable.

4. MODULE DESCRIPTION

A. Feature Selection:

Genetic Algorithms (GAs) are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The main idea of the module is to extract subset. For the taken gene dataset, the individuals in the population are then evaluated and selected. During each successive generation, a proportion of the existing population is selected to breed a new generation. The selection process is done through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. The fitness function is a user defined evaluation function and gives the individuals a score based on how well they perform at the given task.

The individuals are then selected based on their fitness, the higher the fitness, the higher and the chance of being selected. The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation. The selected individuals then "reproduce" to create one or more offspring, after which the offspring are mutated randomly. New parents are selected for each child, and the process continues until a new population of solutions of appropriate size is generated. This continues until a suitable solution has been found or a certain number of generations have passed, depending upon the needs. These processes ultimately result in the next generation population of chromosomes that is different from the initial generation.

B. Classification by KNN:

In this module the gene data are classified using KNN algorithm. K-nearest neighbor is a supervised learning algorithm where the result of new instance query is classified based on majority of K-nearest neighbor category. K Nearest neighbor algorithm used neighborhood classification as the prediction value of the new query instance. The purpose of this algorithm is to classify a new object based on attributes and training samples. The classifiers do not use any model to fit and only based on memory. The selected features are given as an input to this module.

The K (number of nearest neighbors) values are chosen that are closest to the query point. The distance between the query-instance and all the training samples are calculated. The distance are then sorted and nearest neighbors based on the K th minimum distance is determined. The category Y of the nearest neighbors is gathered. The simple majority of the category of nearest neighbors as the prediction value of the query instance is used. Any ties can be broken at random.

C. Classification by SVM:

In this module the gene data are classified using SVM algorithm. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. Viewing input data as two sets of vectors in an n-dimensional space, an SVM will construct a separating hyper plane in that space, one which maximizes the margin between the two data sets. To calculate the margin, two parallel hyper planes are constructed, one on each side of the separating hyper plane, which are "pushed up against" the two data sets.

Intuitively, a good separation is achieved by the hyper plane that has the largest distance to the neighboring data points of both classes, since in general the larger the margin the lower the generalization error of the classifier. The selected features are given as an input to this module. In the case of support vector machines, suppose the incoming data belong to one of two classes, a data point is viewed as a p-dimensional vector (a list of p numbers).There is many hyper planes that might classify the data. The hyper plane is selected based on the distance from the hyper plane to the nearest data point. That is to say that the nearest distance between a point in one separated hyper plane and a point in the other separated hyper plane is maximized.

Work Flow Diagram

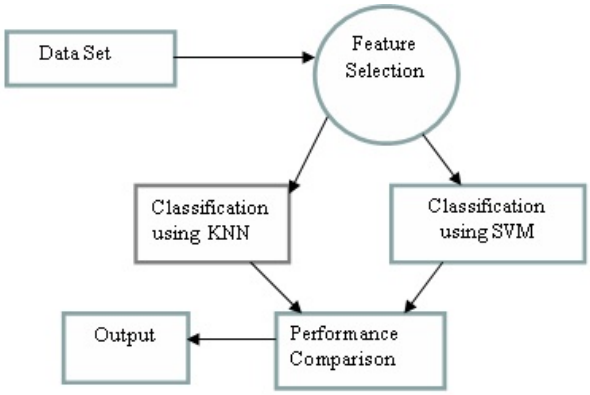


Fig: 1 Work Flow Diagram of the System

5. GENE DATA ANALYSIS USING KNN

In pattern recognition, the ***k*-nearest neighbor algorithm** (KNN) is a method for classifying objects based on closest training examples in the feature space. KNN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. It can also be used for regression.

A. Overview

The ***k*-nearest neighbor algorithm** is amongst the simplest of all machine learning algorithms. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors. *k* is a positive integer, typically small. If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. In binary (two class) classification problems, it is helpful to choose *k* to be an odd number as this avoids tied votes.

The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its *k* nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the

ISSN 2379-1500

nearer neighbors contribute more to the average than the more distant ones.

The neighbors are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. In order to identify neighbors, the objects are represented by position vectors in a multidimensional feature space. It is usual to use the Euclidean distance, though other distance measures, such as the Manhattan distance could in principle be used instead. The *k*-nearest neighbor algorithm is sensitive to the local structure of the data. Hybrid classification algorithm using Differential Evolution (DE) and Least Squares Support Vector Machine (LS-SVM) and LS-SVM technique is used for classification. Since LS-SVM classifier is so sensitive to the changes of its parameter values, DE algorithm is used as an optimization technique for LS-SVM parameters [17].

B. Algorithm

The training examples are vectors in a multidimensional feature space. The space is partitioned into regions by locations and labels of the training samples. A point in the space is assigned to the class *c* if it is the most frequent class label among the *k* nearest training samples. Usually Euclidean distance is used as the distance metric, however this will only work with numerical values. In cases such as text classification another metric, such as the

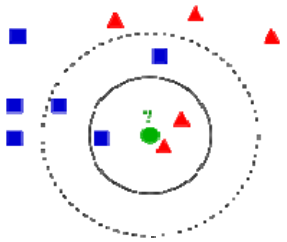


Fig 2: Example for k-NN classification

Overlap metric (or Hamming distance) can be used.

The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If *k = 3* it is classified to the second class because there are 2 triangles and only 1 square inside the inner circle. If *k = 5* it is classified to first class (3 squares vs. 2 triangles inside the outer circle).

The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. In the actual classification phase, the test sample (whose class is not known) is represented as a vector in the feature space. Distances from the new vector to all stored vectors are computed and *k* closest samples are selected. There are a number of ways to classify the new vector to a particular class , one of the most used techniques is to predict the new vector to the most common class amongst the K nearest neighbors. A major drawback to using this technique to classify a new vector to a class is that the classes with the more frequent examples tend to dominate the prediction of the new vector, as they tend to come up in the K nearest neighbors when the neighbors are computed due to their large number. One of the ways to overcome this problem is to take into account the distance of each K nearest neighbors with the new vector that is to be classified and predict the class of the new vector based on these distances.

C. Parameter Selection

The best choice of *k* depends upon the data; generally, larger values of *k* reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good *k* can be selected by various heuristic techniques, for example, cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e. when *k* = 1) is called the nearest neighbor algorithm.

The accuracy of the *k*-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification. A particularly popular approach is the use of evolutionary algorithms to optimize feature scaling. Another popular approach is to scale features by the mutual information of the training data with the training classes.

D. Properties

The naive version of the algorithm is easy to implement by computing the distances from the test sample to all stored vectors, but it is computationally intensive, especially when the size of the training set grows. Many nearest neighbor search algorithms have been proposed over the years; these generally seek to reduce the number of distance evaluations actually performed. Some optimizations involve partitioning the feature space, and only computing distances within specific nearby volumes. Several different types of nearest neighbor finding algorithms include k-Most Similar Neighbor (k-MSN), Linear scan , Kd-trees, Balltrees, Metric trees, Locality sensitive hashing (LSH), Agglomerative-Nearest-Neighbor, Redundant Bit Vectors (RBV).

The *k*-NN algorithm can also be adapted for use in estimating continuous variables. One such implementation uses an inverse distance weighted average of the *k*-nearest multivariate neighbors. This algorithm functions as follows:

1. Compute Euclidean distance from target plot to those that were sampled.
2. Order samples taking for account calculated distances.
3. Choose heuristically optimal *k* nearest neighbor based on root mean square error (RMSE) done by cross validation technique.
4. Calculate an inverse distance weighted average with the *k*-nearest multivariate neighbors.

The following gives the Pseudo code for k-Nearest Neighbor classification with cross-validation:

begin

Initialize the nxn distance matrix D, initialize the Ωx Ω confusion matrix c, set t←0, TotAcc←0, and set NumIterations equal to the desired number of iterations (re-partitions).

Calculate distances between all the input samples and store in nxn matrix D. (For a large number of samples,

International Journal Series

ISSN 2379-1500

use only the lower or upper triangular of D for storage since it is a square symmetric matrix.)

for t←1 **to** NumIterations **do**

Set c←0, and ^{n}total←0.

Partition the input samples into Kequally-sized groups.

for fold←1 **to** k**do**

Assign samples in the f ∞ dth partition to testing, and use the remaining samples for training. Set the number of samples used for testing as ^{n}test.

Set ^{n}total← ^{n}total+ ^{n}test.

for *i* ←1 **to** ^{n}test **do**

For test sample X_i determine the k closest training samples based on the calculated distances.

Determineω, the most frequent class label among the k closest training samples.

Increment confusion matrix C by 1 in element $^{C}\omega,\hat{\omega}$, where ω is the true and $\hat{\omega}$ the predicted class label for test sampleX_i. If $\omega = \hat{\omega}$ then the increment of +1 will occur on the diagonal of the confusion matrix, otherwise, the increment will occur in an off-diagonal.

Determine the classification accuracy using Acc= $\sum_j^{\Omega} C_{jj}$ / ^{n}total where $^{C}_{jj}$ is a diagonal element of the confusion matrix C.

Calculate Tot Acc= TotAcc+Acc.

Calculate AvgAcc = TotAcc/NumIterations..

end

E. Distance Calculation

In mathematics, the **Euclidean distance** or **Euclidean metric** is the "ordinary" distance between two points that one would measure with a ruler, which can be proven by repeated application of the Pythagorean theorem. By using this formula as distance, Euclidean space becomes a metric space (even a Hilbert space). The associated norm is called the **Euclidean norm.**

Older literature refers to this metric as **Pythagorean metric**. The technique has been rediscovered numerous times throughout history, as it is a logical extension of the Pythagorean theorem.

The Euclidean distance between points P = (p_1,p_2,…….p_n) and Q=_q_1,q_2,……..q_n), in Euclidean *n*-space, is defined as:

$$\sqrt{(p_1-q_1)^2+(p_2-q_2)^2+\cdots+(p_n-q_n)^2}=\sqrt{\sum_{i=1}^{n}(p_i-q_i)^2}.$$

Equation No- 1.

F. Classification

Experimental results show that, the subset of data selected using Genetic Algorithm is classified using the KNN. First the Euclidean distance is calculated between all the neighboring data in the dataset. The genetic algorithm has been applied for selecting the best features from a gene expression dataset and KNN and SVM for classification are as followed by the simulation results shown in below.

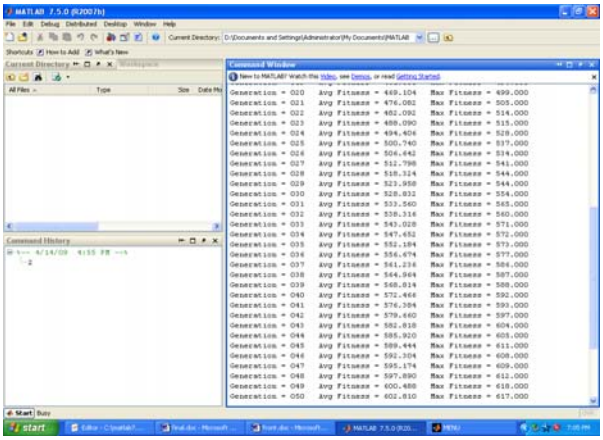
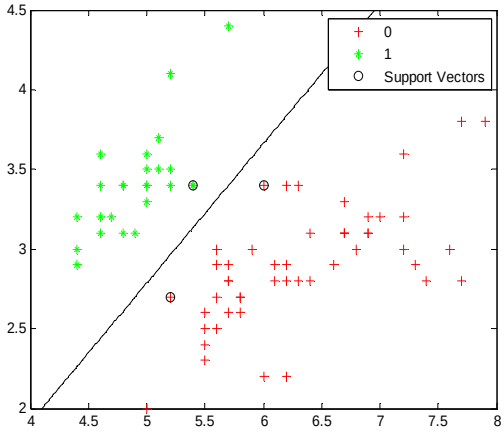


Fig 3. Feature Selection – Average Fitness Values

Classification Vector Values



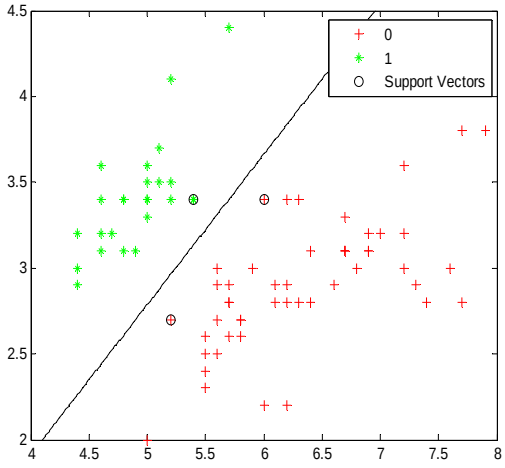


Fig 8. Classification using SVM

International Journal Series

ISSN 2379-1500

6. EXPERIMENTS AND RESULTS

The quick reduct algorithm is an effective way to reduce redundancy. The conditional features can be set depending on the dataset. The KNN uses Euclidean distance formula to calculate neighboring data distance. Here, cross-validation is used in classifying for both KNN and SVM.

To perform KNN it takes a total of 5 seconds whereas it takes 11 seconds for SVM. The following table 2 and Fig 9 shows the experimental results:

Table 2: Experiments and Results

	KN	SV
ErrorRate	0.0	0.1
CorrectRat	0.9	0.9

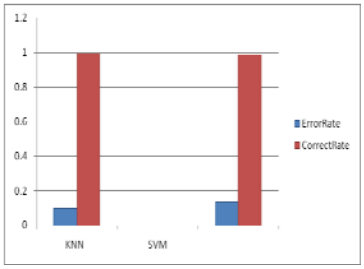


Fig: 9 Experiments and Results

The dataset contains 640 data. After the subset section the dataset contains 462 data. The error rate and correct rate shows that for the given small data set KNN performs better than SVM.

7. CONCLUSIONS

In this paper, the genetic algorithm has been applied for selecting the best features from a gene expression dataset and KNN and SVM is used for classification. Experimental results shows that the better performance. The proposed method was used to prune the mislabeled training samples and can eliminate those samples effectively. Meanwhile the misclassification rates of the proposed method not increase sharply.

In k-NN the major drawback, definition of k, has acted as the reason for achieving a better result compared to a more effective classifier. For this given data set, the KNN gives a better overall performance than SVM though error rate of SVM is lower than that of KNN.

8. FUTURE WORK

The ultimate aim is to develop a better classifier than existing classifiers to classify gene data for both small and large data sets. To achieve this SVM classifier used along with Genetic algorithm is improved by applying boosting techniques like hidden space and alternating decision tree. This is then compared with advanced classifiers like decision tree classifiers and neural networks.

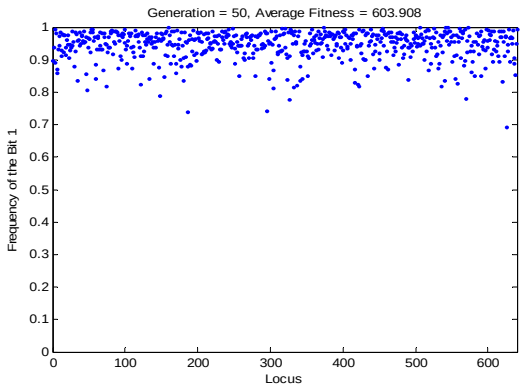


Fig 4. Feature Selection – Frequency of bits against locus

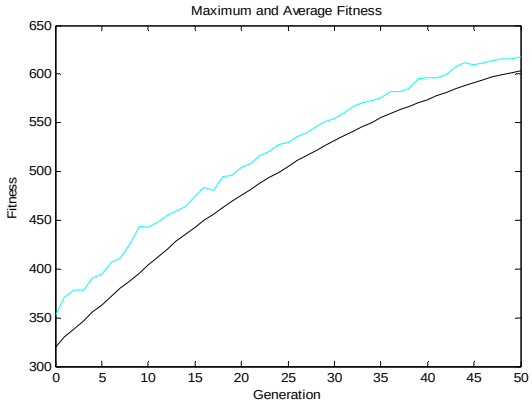


Fig 5. Generation vs. Fitness

Classification using KNN

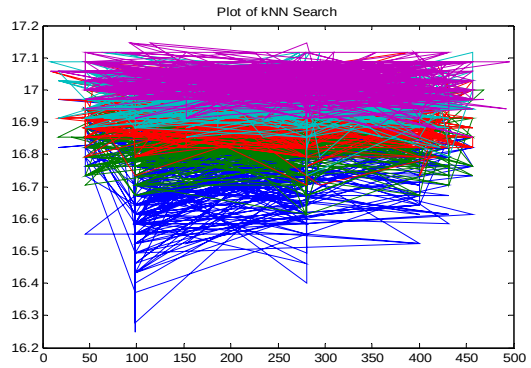


Fig 6. Plot of kNN Search

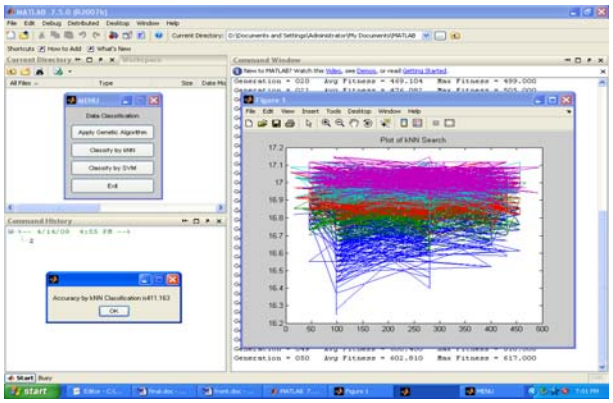


Fig 7. Plot of kNN vs. Max Fitness

ISSN 2379-1500

References :

1. “SVM-based Automatic Classification for Protein Structural Domain” by Xiao-Jian Shao, Ying-Jie Tian, Nai-Yang Deng, The First International Symposium on Optimization and Systems Biology ,2007
2. “Scalable Rule-Based Gene Expression Data Classification (Extended Version)” by Mark A. Iwen, Willis Lang,Jignesh M. Patel,2007
3. “A hybrid structural/statistical classfier for handwritten Farsi/Arabic numerical recognition” by Saeed Mozaffari,Karim faez,Majid Ziaratban,2008
4. “Using a genetic algorithm and a perceptron for feature selection and supervised class learning in DNA micro array data” by Michal Karzynski, Álvaro Mateos, Javier Herrero and Joaquín Dopazo,2003
5. “Markov Blanket-Embedded Genetic Algorithm for Gene Selection” by Zexuan Zhu, Yew-Soon Ong , Manoranjan Dash,2000
6. “Classification of Gene Expression Data by Majority Voting Genetic Programming Classifier” by Topon Kumar Paul, Yoshihiko Hasegawa, and Hitoshi Iba,2002
7. “Aiding Classification of Gene Expression Data with Feature Selection: A Comparative Study” by Changjing Shang and Qiang Shen,2001
8. ”Inferring of genic regulatory networks from expression data using KNN classifiers” by Daniel Nicorici, Sampsa Hautaniemi, Hakan Öktem, Olli Yli-Harja, Jaakko Astola,2008
9. “Gene Selection Using a New Error Bound for Support Vector Machines” by Xin Zhou, David P. Tuck.
10. “A Support Vector Machine Classifier for Gene Name Recognition” by Stefen Bickel, Ulf Brefeld, Lukas Faustich,Jorg Hakenberg, Ulf Leser, Conrad Plake, Tobias Scheler,2004
11. “A Maximally Diversified Multiple Decision Tree Algorithm for Micro array Data Classification” by Hong Hu, Jiuyong Li, Hua Wang, Grant Daggard, Mingren Sh,2006.
12. ”Classification and diagnostic prediction of cancers using Gene Micro Array Data analysis” by Alireza Osareh and Bita Shadgar,2009
13. “Mining housekeeping genes with a Naive Bayes classifier” by Luna De Ferrari,2006
14. “ARC: Automated Resource Classifier for agglomerative functional classification of prokaryotic proteins using annotation texts” by Muthiah Gnanamani, Naveen kumar and Srinivasan Ramachandran,2007
15. “RBNBC: Repeat Based Naive Bayes Classifier for Biological Sequences” by Pratibha Rani,2008
16. “A Genetic Algorithm for Feature Selection in Data-Mining for Genetics” by Laetitia Jourdan, Clarisse Dhaenens, El-Ghazali Talbi,2001
17. “Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms” by Bichen Zheng, Sang Won Yoon, Sarah S. La, 2014.

Biodata for the Authors

Dr. J. Subash Chandra Bose was born in Erode (India) in October 1977. He did his Diploma in Electrical & Electronics Engineering, Bachelor of Engineering in Electrical & Electronics Engineering from Anna University, Chennai, Master of Engineering in Computer Science & Engineering from Anna University, Chennai, Ph. D in Computer Science & Engineering from Anna University, Chennai, India. He is having more than 13 years engineering teaching and Industry experience, also having good inclination towards research in Medical Image Processing, Wireless network Stenography & Cellular Automata Techniques. He has published more than 40 scientific papers in various National and International Journals and Conference proceedings. He has received a **“BEST COMPUTER SCIENCE FACULTY”** Award by Association of Scientists, Developers and Faculties, Bangkok, Thailand. He Acted as a Key note Speaker and Chair for the International Conference on Cryptography and Security 2014 conducted by ASDF, Bangkok, Kingdom of Thailand.

Dr. Jamel Baili, received Ph.D. Degree in 2010 from the university of Monastir, Tunisia. He has 12 years of teaching experience in many universities in Tunisia. He has published about 12 research papers in reputed international journals and conferences. His research interests include Digital signal processing, Instrumentation and Embedded systems. He is currently an Assistant Professor working in Department of Computer Engineering, King Khalid University, Abha, KSA

Dr.Ahmed Said Badawy received Ph.D in Communication Engineering from Alexandria University, Egypt. He has 21 years of teaching experience. His research areas of interest include Digital Signal Processing, Signal separation, CDMA and Embedded systems. He is currently Assistant Professor in Dept of Computer Engineering, College of Computer Science, King Khalid University, Abha, Saudi Arabia

Dr. Wade Ghribi received Ph.D degree from Computer Engineering Faculty, Kharkov National University of Radio electronics, Kharkov, Ukraine. He is currently Assistant Professor in Dept of Computer Engineering, College of Computer Science, King Khalid University, Abha, Saudi Arabia.

Dr.Suresh Babu Changalasetty received Ph.D in Computer Science and Engineering from Acharya Nagarjuna Univeristy, Guntur, India. He has 15 years of teaching experience. His research interests are bioinformatics, image processing and data mining. He is currently Associate Professor in Dept of Computer Engineering, College of Computer Science, King Khalid University, Abha, Saudi Arabia.

Mr. Harun Bangali .MTech in Electronics and Communication .He has 11 years of teaching experience. He is currently Faculty in Dept of Computer Engineering, College of Computer Science, King Khalid University, Abha, Saudi Arabia.

ISSN 2379-1500