Comparitive Analysis Of Classification Algorithm In Multiple Categories Of Bioinformatics

D.CHANDRA VARMA (M.TECH)
VIGNAN'S INSTITUTE OF INFORMATION AND TECHNOLOGY,VISAKHAPATANAM.
DEPT OF COMPUTER SCIENCE & ENGINEERING

D.DHARMAIAH M.TECH,(Ph.D)
VIGNAN'S INSTITUTE OF INFORMATON AND TECHNOLOGY,VISAKHAPATANAM.
DEPT OF COMPUTER SCIENCE & ENGINEERING

Abstract

This article reviews machine learning methods for bioinformatics. It presents modellingmethods, such as supervised classification, clustering as well as deterministic and stochastic heuristics for optimization. Applications in genomics, proteomics and text mining are also shown. In this process by using the algorithms like Bayesian classification, Back propagation neural network and Support vector machine to find the people effected from the cancer (or) not. This test will be done in the genomics for testing purpose leukemia is taken which was present in the red blood cells, the data will be taken from the // ds: http://www.upo.es/eps/bigs/datasets.html by applying the leukemia in the algorithms which mentioned above. We can find that which type of algorithm is user free and in which process result will come accurately. While coming to the proteomics which type of structure better for the process can be seen.

KEYWORDS:

Machine learning, Supervised classification, Genomics, Proteomics, Bayesian algorithm, Back propagation neural network, Support vector machines.

1. INTRODUCTION

Machine learning consists in programming computers to optimize performance criteria on by using example data or past experience. The optimized criterion can be the accuracy provided by a predictive model in a modeling problem and the value of a fitness or evaluation function in an optimization problem.

The 'learning' term refers to running a computer program to induce a model by using training data or past experience. Machine learning uses statistical theory when building computational models since the objective is to make inferences from a sample. The two main steps in this process are to induce the model by processing the huge amount of data and to represent the model and making inferences efficiently. It must be noticed that the efficiency of the learning and inference algorithms, as well as their space and time complexity and their transparency and interpretability, can be as important as their predictive accuracy. The

process of transforming data into knowledge is both iterative and interactive. The iterative phase consists of several steps.

Step I :- We need to integrate and merge the different sources of information into only one format. By using data warehouse techniques, the detection and Resolution of outliers and inconsistencies are solved.

Step II :- It is necessary to select, clean and transform the data. To carry out this step, we need to Eliminate or correct the uncorrected data, also selects the relevant and non-redundant variables. This is done with respect to the instances.

Step III :- The data mining, we take the objectives of the study into account in order to choose the most appropriate analysis for the data.

In this step, the type of paradigm for supervised or unsupervised classification should be selected and the model will be induced from the data. Once the model is obtained, it should be evaluated and interpreted both from statistical and biological points of view and, if necessary, we should return to the previous steps for a new iteration. This includes the solution of conflicts with the current knowledge in the domain. The model satisfactorily checked and the new knowledge discovered .This is used to solve the problem. Optimization problems can be posed as the task of

Finding an optimal solution in a space of multiple(sometimes exponentially sized) possible solutions. The choice of the optimization method to be used is crucial for the problem solution. Optimization approaches to biological problems can be classified, according to the type of solutions found, into exact and approximate methods. Exact methods output the optimal solutions when convergence is achieved. However, they do not necessarily converge for every Instance.

Approximate algorithms always output a candidate solution, but it is not guaranteed to be the optimal one. Optimization is also a fundamental task when modelling from data. In fact, the process of learning from data can be regarded as searching for the model that gives the data the best fitting.

2. SYSTEM ANALYSIS

The analysis of the requirements make it clear that the user needs a user friendly graphical user interface.So,we have used NetBeans and written the code in Java language.

All the event handlers for every control on the form wherever any processing is required is provided. All the code is placed within try and catch so as to apply error handlers and a suitable message is displayed in case of errors.

The purpose of the requirement capture analysis is to aim the development toward the right system. Its goal is to produce a document called requirement specification. The whole scope of requirement capture and analysis forms the so-called requirement engineering.

The document of the requirement specification will be used as

- The agreed contract between the client and the system development organization on what the system should do (and what the system should not do);
- The basis used by the development team to develop the system;

- A fairly full model of what is required of the system.

  To fulfill these purposes, the requirement analysis process, should include the following highly iterative activities

3. DOMAIN ANALYSIS

Domain understanding Analysts must develop their understanding of the application domain. Therefore, the concepts are explored and the client's requirements are elicited.

- **Domain Analysis document for Machine learning in Bioinformatics**

Introduction

The system is classified into several methods for the classification of the problems by the six domains ;

There are Genomics, proteomics, Microarray, Systems Biology, Evolution and TextMining.

Genomics :-

Genomics is one of the most important domains in bioinformatics. The number of sequences available is increasing exponentially. These data need to be processed in order to obtain useful information from genome sequences; we can extract the location and structure of the genes. If the genes contain the information, proteins are the workers that transform this information into life. Proteins play a very important role in the life process,

Proteomic :-

The main application of computational methods is protein structure prediction. Proteins are very Complex macromolecules with thousands of atoms and bounds. Hence, the number of possible structures is huge. This makes protein structure prediction a very complicated combinatorial problem where optimization techniques are required in proteomics.

Text Mining :-

Text Mining techniques are required for the knowledge extraction. Thus, text mining is becoming more and more interesting in computational biology, and it is being applied in functional annotation, cellular location prediction and protein interaction analysis.

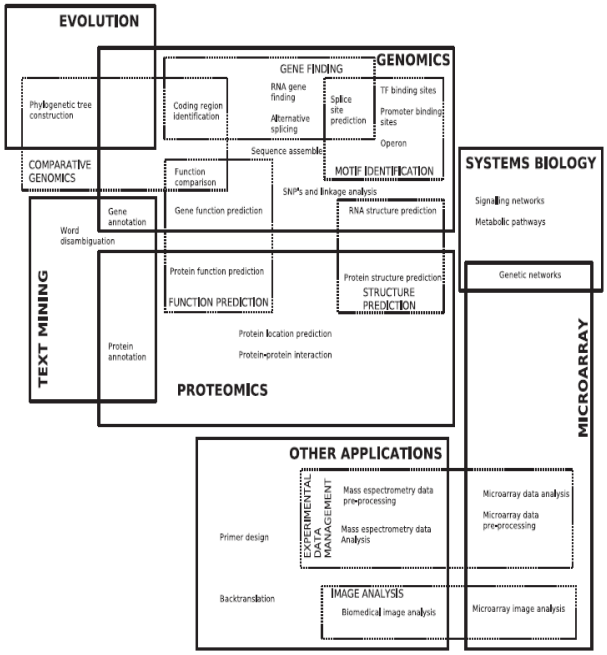

FIG: The view of machine learning Bioinformatics

4. SCOPE OF THE SYSTEM

The past decade has seen tremendous growth in the amount of biomedical data. In particular, the sequencing of the human genome and of quite a few other organisms has generated complete genomic sequences of unprecedented number and size. This development is accompanied by much data of various kinds, including protein sequences, results from large-scale genomic

and proteomic experiments, and a lot of published literature. Recent research in biomedical computing and informatics is focused on the interpretation of such data. The ultimate goal is to understand and predict normal function of organisms and, more importantly, the mechanisms underlying disease. Given the wealth of data - the interpretation cannot be done manually. It requires advanced computational tools, mimicking some aspects of the manual interpretation.

Machine Learning is concerned with the automatic acquisition of models from data, as well as with the usage of such models for automatic inference and prediction. As addressing biomedical data interpretation is concerned both with modeling complex biological systems, and with deducing/predicting the role of genes and proteins within these systems, machine learning methods have much to offer, and are currently applied to a wide variety of biomedical problems. In this seminar we'll survey methods and approaches in machine learning, along with current applications in biomedical informatics.

5. PROBLEM STATEMENT

In recent years, genomics has increased the understanding of many diseases. Proteomics is a rapidly growing research area that encompasses both genetic and environmental factors. The protein composition represents the functional status of a biological compartment. The five approaches presented here resulted in the detection of disease-associated proteins. Calgranulin B was unregulated in colorectal cancer, and hepatoma-derived aldose reductase-like protein was reexpressed in a model during hepatocarcinogenesis. In these two investigations, attention was focused on one protein, obviously differing in amount, directly after two-dimensional elctro phoresis(2-DE)

2-DE database was constructed, containing 3300 protein spots and 150 identified protein species. The number of identified proteins was limited by the capacity of our group, rather than by the principle of feasibility. Another field where proteomics proves to be a valuable tool in identifying proteins of importance for diagnosis is proteome analysis of pathogenic microorganisms such as Borrelia burgdorferi (Lyme disease) and Toxoplasma gondii (toxoplasmosis). Sera from patients with early or late symptoms of Lyme borreliosis contained antibodies of various classes against about 80 antigens each, containing the already described antigens OspA, B and C, flagellin, p83/100, and p39. Similarly, antibody reactivity to seven different marker antigens of T. gondii allowed differentiation between acute and latent toxoplasmosis, an important diagnostic tool in both pregnancy and immunosuppressed patients.

6. IMPLEMENTATION

Algorithm:

(i)The Back Propagation learning:

A back propagation neural network uses a feed-forward topology, supervised learning, and back propagation learning algorithm. This algorithm was responsible in large part for the re-emergence of neural networks in the mid 1980s. Back propagation is a general purpose learning algorithm. It is powerful but also expensive in terms of computational requirements for training. A back propagation network with a

single hidden layer of processing elements can model any continuous function to any degree of accuracy (given enough processing elements in the hidden layer). The basic back propagation algorithm consists of three steps.

Step I :-The input pattern is presented to the input layer of the network. These inputs are propagated through the network until they reach the output units. This forward pass produces the actual or predicted output pattern.

Step II:-Because back propagation is a supervised learning algorithm, the desired outputs are given as part of the training vector. The actual network outputs are subtracted from the desired outputs and an error signal is produced.

Step III :-This error signal is then the basis for the back propagation step, whereby the errors are passed back through the neural network by computing the contribution of each hidden processing unit and deriving the corresponding adjustment needed to produce the correct output. The connection weights are then adjusted and the neural network has just "learned" from an experience.

Back propagation Algorithm

First apply the inputs to the network and work out the output – remember this initial output could be anything, as the initial weights were random numbers.

Perform Summation → apply sigmoid function →(1/1+e^-x)

1.Next work out the error for neuron B. The error is What you want – What you actually get, in other words: ErrorB = OutputB (1-OutputB)(TargetB – OutputB)

The "Output(1-Output)" term is necessary in the equation because of the Sigmoid Function – if we were only using a threshold neuron it would just be (Target – Output).

2. Change the weight. Let W+AB be the new (trained) weight and WAB be the initial weight.

W+AB = WAB +Learning rate (ErrorB x OutputA)

Notice that it is the output of the connecting neuron (neuron A) we use (not B). We update all the weights in the output layer in this way.

3.Calculate the Errors for the hidden layer neurons. Unlike the output layer we can't calculate these directly (because we don't have a Target), so we Back Propagate them from the output layer (hence the name of the algorithm). This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors. For example if neuron A is connected as shown to B and C then we take the errors from B and C to generate an error for A.

ErrorA = Output A (1 - Output A) (ErrorB WAB + ErrorC WAC)

Again, the factor "Output (1 - Output)" is present because of the sigmoid squashing function.

4. Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method we can train a network of any number of layers.

(ii)Bayesian classifiers:

Bayesian classifier minimize the total misclassification cost using the following assignment:

$$\gamma(x) = \arg\min k \sum_{c=1}^{r0} cost(k,c)p(c|x1,x2,..xn)$$

where cost(k, c) denotes the cost for a bad classification. In the case of a 0/1 loss function, the Bayesian classifier assigns the most probable a posteriori class to a given instance, that is:

$\boldsymbol{\gamma}$(x) = arg max c p(c|x1, x2, . . . , xn) = arg maxc p(c)p(x1, x2, . . . , xn|c).

Depending on the way p(x1, x2, …. xn|c) is approximated, different Bayesian classifiers of different complexity are obtained.

Naive Bayes is the simplest Bayesian classifier. It is built upon the assumption of conditional independence of the predictive variables given the class. Although this assumption is violated in numerous occasions in real domains, the paradigm still performs well in many situations. The most probable a posteriori assignment of the class variable is calculated as

$C^* = \arg\max_c p(c|x_1,x_2,\ldots,x_n) = \arg\max_c p(c)\prod_{i=1}^{n} p(Xi|c)$

The seminaiveBayes classifier tries to avoid the assumptions of conditional independence of the predictive variables, given the class variable, by taking into account new variables. These new variables consist of the values of the Cartesian product of domain variables which overcome a condition. Independence concept and the reliability on the conditional probability estimations. The tree augmented naive Bayes classifier also takes into account relationships between the predictive variables by extending a naive Bayes structure with a tree structure among the predictive variables. This tree structure is obtained adapting the algorithm proposed by Chow and Liu and calculating the conditional mutual information for each pair of predictive variables, given the class.

(iii)Support vector machines:

Support vector machines rely on pre-processing the data to represent patterns in a high dimension— typically much higher than the original feature space. With an appropriate non-linear mapping to a sufficiently high dimension, data from two categories can always be separated by a hyperplane. This choice will often be informed by the designer's knowledge of the problem domain. In absence of such information, one might choose to use polynomials,

Gaussians, or other basic functions. The dimensionality of the mapped space can be arbitrarily high (though in practice it may be limited by computational resources). Defining the margin as any positive distance from the decision hyperplane, the goal in training support vector machines is to find the separating hyperplane with the largest margin. We expect that the larger the margin, the better the generalization of the classifier

7. TEST PLAN AND TEST CASE

Test No.	Test Case	Expected Output	Actual Output	Result
1.	Input & output Files:	Only CSV files are	Only CSV files are	Passed

	only coma Separated values files are allowed	accepted as input& Output files	accepted as input Output files	
Test No.	**Test Case**	**Expected Output**	**Actual Output**	**Result**
2.	Maximum n.o of Epochs: N.o of epochs the user uses must be limited to particular value	Maximum n.o of epochs are limited to 500	Maximum n.o of epochs are limited to 500	Passed
3.	Validate input file:The first row of an input file is not considered as it contains the attributes	Values are read from the second row of an input file	Values are read from the second row of an input file	Passed
4.	Validate output file:The first row of the output file is considered as the result	The first row of an output file is read as an input for post processing	The first row of an output file is read as an input for post process	Passed

8. CONCLUSION

Nowadays, one of the most challenging problems in computational biology is to transform the huge volume of data, provided by newly developed technologies, into knowledge. Machine learning has become an important tool to carry out this transformation.

This article introduces some of the most useful techniques for modelling—Bayesian classifiers, logistic regression, discriminate analysis, classification trees, nearest neighbor, neural networks, support vector machines, ensembles of classifiers, partitional clustering, hierarchical clustering, mixture models, hidden Markov models, Bayesian networks and Gaussian networks and optimization Monte Carlo algorithms, simulated annealing, tabu search, GAs, genetic programming and estimation of distribution algorithms—giving some pointers to the most relevant applications of the former techniques in bioinformatics.

9. References:

1. Mathé C, Sagot M.-F, Schlex T, et al.Current methods of gene prediction, their strengths and weaknesses. Nucleic Acids Research 2002;30(19):4103–17.
2. Stein Aerts, Peter Van Loo, Yves Moreau, et al. A genetic algorithm for the detection of new cis-regulatory modules in sets of coregulated genes. Bioinformatics 2004;20(12): 1974–76.
3. Bock horst J, Craven M, Page D, et al. A Bayesian network approach to operon prediction. Bioinformatics 2003;19(10): 1227–35.
4. Won K.-J, Prü gel-Bennet A, Krogh A. Training HMM structure with genetic algorithm for biological sequence analysis. Bioinformatics 2004;20(18):3613–19.
5. Carter RJ, Dubchak I, Holbrook SR. A computational approach to identify genes for functional RNAs in genomic sequence. Nucleic Acids Research 2001;29(19): 3928–38.
6. Durbin R, Eddy SR, Krogh A, et al. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, 1998.
7. Gary B. Fogel, David W. Corne. Evolutionary Computation in Bioinformatics. Morgan Kaufmann, 2002.
8. Frasconi P, Shamir R (eds). Artificial Intelligence and Heuristic Methods in Bioinformatics, Volume 183, NATO Science Series: Computer and Systems Sciences Edited. NATO, 2003.
9. Higgins D, Taylor W (eds). Bioinformatics. Sequence, Structure, and Databanks. Oxford University Press, 2000.
10. Husmeier D, Dybowski R, Roberts S (eds). Probabilistic Modeling in Bioinformatics and Medical Informatics. Springer Verlag, 2005.