 Proceedings of ICCES'08, pp.139-146

VB Classif 1.0 a Novel Tool to Classify the Epidemiological Data of Vector Borne Diseases

U.Suryanarayana Murty[1] and V.Sreehari Rao[2]

Summary

Artificial intelligence has opened a new window of opportunity and has definitely a long way to look ahead in area of data analysis. The enormity of magnitude of data and its complex nature often perplex the epidemiologists. This is especially true in case of vector borne diseases where a timely and precise understanding of disease during decision making process comes handy for curbing the diseases in event of an outbreak or epidemic. Often the public health officials feel the need of correct identification of true positive cases. A tool that classify disease according to presence or absence of a disease will help in devising a clear strategy in mass drug administration programmes and will help us in proper targeting of patients and in efficient use of resources. Available and well known statistical tools tend to settle for a compromise among accuracy, speed and efficiency. Interactive Classification tools supported by AI (Artificial Intelligence) like VB Classif 1.0 will definitely pave way for more efficient disease control and help epidemiologists in finding quick solution to classification problems.

Introduction

"*Crude classification and false generalizations are the curse of organized life*." But that does not halt us from classifying objects and extending these generalizations to a vast category. Scientists have been classifying objects since time immemorial and this effort to group the objects so that they fit nicely together with other similar objects has led to simplification and organization of otherwise chaotic world. Classification in Biology dates back to Aristotle's time and has now reached such sophistication that human effort is often minimized and its place has been superseded by state of art computer- aided tools. With the accelerating advances in high throughput methods generation of data is increasing exponentially. Astonishingly high rate of data generation and flow of data has not kept pace with the development of accurate and precise classification tools that can bridge the gap between the demand and supply. In the Biological field, classification tools are used vigorously in the taxonomical classification to emulate the ambiguities in assigning the appropriate rank [1, 2]. Several applications can be found with classification tools like gene expression analysis [3, 4] structural classification of proteins (SCOP) [5], classification of methylation array data and protein proteomic analysis for cancer [6, 7], for elucidating the evolutionary distance among several species , biological sequences comparison [8], protein structure prediction [9]

[1]Biology Division, Indian Institute of Chemical Technology, Hyderabad-500007, India
[2]Dept Mathematics, JNTU, Hyderabad, India

assigning scores to the drug molecules in docking studies, in ranking the small molecules based on its structure activity relationship (QSAR) [10] and structure property relationship (QSPR). These classification tools are mainly based on hard-core statistics and mathematical models. Principal component analysis (PCA), Partial least square method and regression analysis methods lie in the heart of these tools. Among the myriad of classification tools, HCA (Hierarchical Cluster Analysis),MDS (Multi Dimensional Scaling), CART (Classification And Regression Tree), FP (Frequency Pattern) Tree, SOM (Self Organizing Maps), Correlation coefficient clustering ,SVM (Support Vector Machine), BP-ANN (Back Propagation Artificial Neural Network), GA (Genetic Algorithm), Genetic Programming ,K-Means and K-Nearest neighbor are to name a few.

Vector borne diseases often tends to be complicated and enormous data generating from the experimental studies warrant need of efficient tools for analysis. Though computer technology is undergoing a revolution yet its application in classification is still in infancy especially its potential has not been tapped in arena of vector borne diseases. Often magnitude of epidemiological data and its vast array of types poses a challenge for an epidemiologist. In our gold rush to information, we end up settling to a compromise between accuracy and speed. One of the common statistical tools used in these kinds of studies is K nearest neighbor.

K nearest neighbor

The *k*-nearest neighbor algorithm is a simple instance-based learning method for performing general, non-parametric classification. First introduced by the researchers E. Fix and J. Hodges in their paper, "Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties", in 1951, it is well explored in the literature and has been shown to have good classification (prediction) performance on a wide range of real world data sets (Xiong H. and Chen XW. 2006). It is simple and straight forward to implement. KNN is based on a distance function for pairs of observations, such as the Euclidean distance. In this classification paradigm, k nearest neighbors of a training data is computed first. Then the similarities of one sample from testing data to the k nearest neighbors are aggregated according to the class of the neighbors, and the testing sample is assigned to the most similar class. The performance of the kNN algorithm is influenced by three main factors: (1) the distance metric used to locate the nearest neighbors; (2) the decision rule used to derive a classification from the k-nearest neighbors; and (3) the number of neighbors used to classify the new sample. kNN classifiers are well-suited to solve the given problem because they do not have to spend additional effort for distinguishing additional classes. One of advantages of KNN is that it is well suited for multi-modal classes as its classification decision is based on a small neighborhood of similar objects. So, even if the target class is multi-modal (i.e. consists of objects

whose independent variables have different characteristics for different subsets), it can still lead to good accuracy. Some pleasant aspects of the nearest neighbor (NN) classifier: (1) Many other techniques (such as decision trees and linear discriminants require the explicit construction of a feature space, which for some distance functions is intractable (2) The NN classifier deals with the hugely multiclass nature of visual object recognition effortlessly. (3)From a theoretical point of view, it has the remarkable property that under very mild conditions, the error rate of a KNN classifier tends to the Bayes optimal as the sample size tends to infinity. (4). Additionally, kNN classifiers can be applied to any type of object representation as long as a distance measure is available. Unfortunately, kNN classification has a major drawback as well. The efficiency of classification is rapidly decreasing with the number of training objects. k Nearest Neighbor Classifier. A major drawback of the similarity measure used in KNN is that it uses all features equally in computing similarities. This can lead to poor similarity measures and classification errors, when only a small subset of the features is useful for classification. The accuracy of the k-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the features scales are not consistent with their relevance. A main weakness of this technique is that performance is subject to the often *ad hoc* choice of similarity metric, especially for heterogeneous datasets from which the derived features are of different types and scales, and are correlated. In addition, the standard KNN methods suffer from the curse of dimensionality. The neighborhood of a given point becomes very sparse in a high dimensional space, resulting in high variance. The algorithm is easy to implement, but it is computationally intensive, especially when the size of the training set grows there can be no tie. These resulting class labels are used to classify each data point in the data set. But due to shortcomings faced by kNN, an imperative need was felt for a more robust tool with high efficiency and accuracy.

Based on the kNN approach, a novel tool VB Classif ver.1.0 (fig-1) for classification of epidemiological data of vector-borne diseases was developed.

VB Classif is a novel efficient classification algorithm (designated as VB Classif 1.0), which classifies the records of those affected by Filariasis. This software has been utilized to classify up to a 100000 records and the effective classification yield percentage is 94%.

Dataset

A real life dataset amounting to data of size of 5602 records pertaining to a major disabling vector – borne disease Filariasis was used for this study. Epidemiological, socio-economic and entomological data was collected from East and West Godavari districts of Andhra Pradesh and considered for this study.

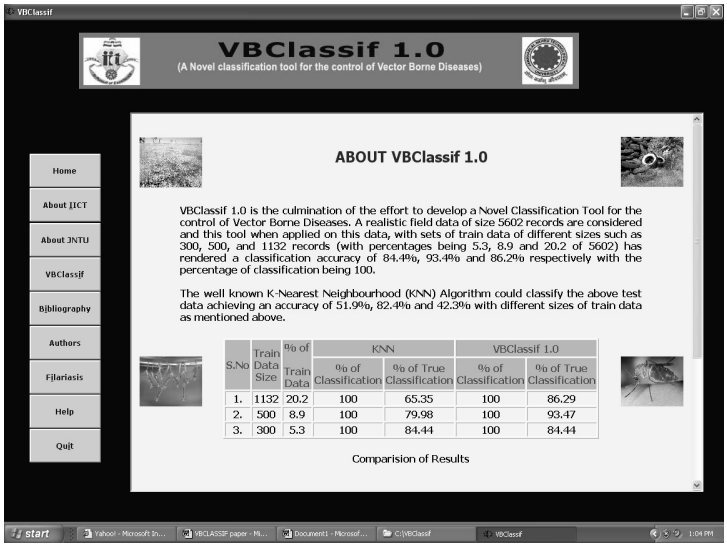

Figure 1: Snapshot showing VB Classif 1.0 application window.

Data Normalization

The data was normalized and each record was assigned a value of 0 or 1, based on presence or absence of the disease characteristics.

Methodology

The data on various parameters like age, sex, and class etc. of different individuals affected by Filariasis is considered. The program is trained with a small portion of data designated as *train data* and is then tested for efficiency by executing on *test data*, which is expectedly large. The algorithm is tested with minimal train data and observed that the efficiency either increases or remains unaltered.

Algorithm

From the entire given test data

1. Separate '0' class records and '1' class records in train data.
2. Take a record from test data and calculate Eucledian distance for each record in train data for both the classes separately.
3. Take least distance in '0' class and '1' class.
4. Compare the least distance in both classes, put the query record in that class which is having least distance.
5. If distances are equal those are taken as unclassified records.
6. Repeat the step 2 to step 5 for every record in test data.

Now, on the unclassified records from the above stage perform following steps

7. Take the least distance record (say r1) in '0' class and least distance record (say r2) in '1' class records that are classified.
8. Take the record in unclassified list, which is having minimum distance (say u1).
9. If u1 is less than both r1 and r2 than take minimum of r1 and r2.
10. If u1 is less than r1 than assign u1 to '0' class. If u1 is less than r2 than assign u1 to '1' class
11. Repeat the steps 3 and 4 for all unclassified records from the I method.

Again, on the unclassified records from the above stage perform following steps

12. Calculate least (L1, L2) and highest (H1, H2) distances in '0' and '1' classes respectively.
13. Calculate average 'r' as
$$r = (H1 + L2)/2 \tag{1}$$
14. Calculate the distance (d1 and d2) between a record in unclassified list and the record that is having H1 distance and L2 distance.
15. If d1 < r then assign query record to '0' class otherwise assign query record to '1'.
16. Repeat the steps from 1 to 4 for each record in unclassified list and also for true negative list.

On the unclassified list perform the following operations

17. Calculate the distances between a record in unclassified list and remaining records in the same list.
$$(x_1^-, x_2, x_3, x_4, x_5, \ldots, x_n) \text{ are records in unclassified list.} \tag{2}$$
$$\text{Calculate the distances } ((x_1, x_2), (x_1, x_3), (x_1, x_4), \ldots, (x_1, x_n)). \tag{3}$$
18. Take the least distance (d1) from these calculated distances.
19. Take least distances (r1, r2) in and '0' and '1' classes resp.
20. Compare d1 with the minimum of r1 and r2 (say m1).
21. Assign d1 to the class to which m1 belongs.
22. Repeat step 1 to step 5 for every record in unclassified list and also in True Negatives list.

Again on the unclassified list perform the following steps

23. Calculate the Eucledian distance between origin and a record in unclassified list.

    If $(x1, x2, x3, \ldots xn)$ are fields in a record then

$$R1 = \text{sqrt}((x1)^2 + (x2)^2 + (x3)^2 + \ldots + (xn)^2) \tag{4}$$

24. Divide every field value by R1

$$(x1/R1, x2/R1, x3/R1, \ldots, Xn/R1) \tag{5}$$

25. Repeat the steps 1 and 2 for the new field values i.e.

$$(x1/R1, x2/R1, x3/R1, \ldots, Xn/R1) \tag{6}$$

26. Calculate the Eucledian distance between a record with new field values in unclassified list and every record in classified records of '0' and '1' classes.
27. Take least distance in '0' class and '1' class.
28. Compare the least distance in both classes, put the query record in that class which is having least distance.
29. Repeat the steps from 1 to 6 for every record in unclassified list and True negative list.

Table 1: The results obtained with different magnitudes of train data

S.No	Total records	Number of Test Data Records	Number of Train Data Records	% of Train Data	% of True Classification
1.	4470	3370	1100	25	83.38
2.	4470	3870	600	13.5	92.77
3.	4470	3970	500	12	91.3
4.	4470	4170	300	6.7	98.4

The algorithm has performed effectively compared to other well known classification algorithms such as K-Nearest Neighborhood (KNN) algorithm. The comparison results are tabulated in Table 1.

Performance evaluation

The results in Table 1 help one to conclude that it is not necessary to have always a large data records for training purpose and what is important is always the minimal set of useful data records. Clearly, a set of 300 train data records could capture the full information and render most effective classification and prediction.

When applied to data with sets of train data of different sizes such as 300, 500, 600 and 1100 records (with percentages being 6.7,12,13.5, 25 of 4470, VB Classif

Table 2: The comparison results of KNN with VB Classif

S. No	Total records	Number of Test Data Records	Number of Train Data Records	% of Train Data	% of True Classification *(KNN)*	% of True Classification *(VB Classif 1.0)*
1.	4470	3370	1100	25	81.72	83.38
2.	4470	3870	600	13.5	87.26	92.77
3.	4470	3970	500	12	92.42	91.30
4.	4470	4170	300	6.7	96.82	98.40

ver 1.0 has rendered a classification accuracy of 98.4,91.3,92.77,83.38 respectively with the percentage of classification being 100 while the well known K-Nearest Neighborhood (kNN) Algorithm could classify the above test data achieving an accuracy of 96.82, 92.42, 87.26 and 81.72 with different sizes of train data as mentioned above.

The tool demonstrates high percentage of classification accuracy as compared to the well known kNN, which is a desired feature. VB classif is a menu- driven and user friendly, robust tool which even a novice can apply to analyse the vast amount of data. This tool offers an advantage of being applicable to two-tier classification environments. This tool can be extrapolated to any vector- borne diseases and hence, provide an effective way of emulating the disease.

Acknowledgement

Authors are grateful to Director IICT Hyderabad for encouragement and support. Thanks are due to Department of Biotechnology, Govt. of India for their financial support.

References

1. **Hyun-Jin, C.S.; Timo, A.N.; Mark, B.; Nan, B.** (2002): Improving Behaviour Classification Consistency: A Technique from Biological Taxonomy, *Australian Association for Research in Education Annual Conference*, CHO02101.

2. **Graham, M.; Kennedy, J.B.; Hand, C.** (1999): The challenge of visualising multiple overlapping classification hierarchies, *Proc User Interfaces to Data Intensive Systems*, vol. 6, pp.42–51.

3. **Wang, D.; Lv, Y.; Guo, Z.; Li, X.; Li, Y.; Zhu, J.; Yang, D.; Xu, J.; Wang, C.; Rao, S.; Yang, B.** (2006): Effects of replacing the unreliable cDNA microarray measurements on the disease classification based on gene expression profiles and functional modules, *Bioinformatics*, vol. 29 (ahead of print).

4. **Xiong, H.; Chen, X.W.** (2006): "Kernel-Based Distance Metric Learning for Microarray Data Classification, *BMC Bioinformatics*, vol. 7, pp. 299.

5. **Murzin, A.G.; Brenner, S.E.; Hubbard, T.; Chothia, C.** (1995): SCOP-a structural classification of proteins database for the investigation of sequences and structures, *Journal of Molecular Biology*, Vol. 247, pp. 536-540.

6. **Model, F.; Adorjan, P.; Olek, A.; Piepenbrock, C.** (2001): Feature selection for DNA methylation based cancer classification, *Bioinformatics*, vol. 17, pp.157-164.

7. **Ratsch, G.; Sonnenburg, S.; Schafer, C.** (2006): Learning interpretable SVMs for biological sequence classification, *BMC Bioinformatics*, vol. 7 (Suppl 1) S9.

8. **Orengo, C.A.; Michie, A.D.; Jones, S.; Jones, D.T.; Swindells, M.B.; Thornton, J.M.A** (1997): Hierarchic classification of protein domain structures", *Structure*, vol. 5, pp.1093-1108.

9. **Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J.C.; Sheridan, R.P.; Feuston, B.P.** (2003): Random forest: a classification and regression tool for compound classification and QSAR modeling, *Chem Inf Comput Sci*, vol. 43, pp.1947-1958.

10. **Tom, M.M.** (1997): *Machine Learning*. McGraw-Hill, 1997.