A Novel Tool for Classification of Epidemiological Data of Vector-Borne Diseases

Sree Hari Rao Vadrevu, Suryanarayana U Murty

Department of Mathematics and Statistics 202, Rolla Building 400 W 12th Street ROLLA USA, Biology Division, Indian Institute of Chemical Technology, Hyderabad - 500 018, India

ABSTRACT

In this article we present a novel tool that renders efficient classification of epidemiological data of vector-borne diseases. This algorithm has been applied on the data of the Filariasis disease and the results are compared with the well-known k-nearest neighbor algorithm.

Key words: Epidemiology, Vector-borne disease, Filariasis, k-nearest neighbor algorithm, VBClassif

INTRODUCTION

Crude classification and false generalizations are the curse of organized life. However, 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 the simplification and organization of an 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to Aristotle’s time and has now reached such sophistication that human effort is often minimized and its place has been superseded by state-of-the-art computer-aided tools. With the accelerating advances in high through put methods, the generation of data is increasing exponentially. An 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 demand and supply. In the biological field, classification tools are used vigorously in taxonomical classification to emulate the ambiguities in assigning the appropriate rank. Several applications can be found with classification tools, such as, gene expression analysis, structural classification of proteins (SCOP), classification of methylation array data, and protein proteomic analysis for cancer for elucidating the evolutionary distance among several species, comparison of biological sequences, protein structure prediction, assigning scores to drug molecules in docking studies, and in ranking the small molecules based on their structure activity relationship (QSAR) and structure property relationship (QSPR). These classification tools are mainly based on hardcore statistics and mathematical models. Principal component analysis (PCA), partial least square method, and the regression analysis method 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, and k-Means and k-nearest neighbor (kNN) are some of them. Vector-borne diseases often tend to be complicated and the enormous data generating from the experimental studies warrant a need for efficient tools, for analysis. Although computer technology is undergoing a revolution its application in classification is still in its infancy, its potential, especially, has not been tapped in the arena of vector-borne diseases. Often, the 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 for a compromise between accuracy and speed. One of the common statistical tools used in these kinds of studies is the kNN.

The present article is organized as follows: In Section 2, we describe the kNN approach and in Section 3 we present the methodology, the data sets, and data normalization. The main algorithm is discussed in Section 4. The results and performance evaluation of the novel algorithm form the content of Section 5, while the conclusions are deferred to Section 6.

Address for correspondence:
Prof. Sree Hari Rao Vadrevu, E-mail: vshrao@gmail.com
Nearest neighbor

The kNN algorithm is a simple instance-based learning method for performing general, nonparametric classification. First introduced by the researchers E. Fix and J. Hodges in their article, ‘Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties’, in 1951, it is well explored in literature and has been seen to have a good classification (prediction) performance on a wide range of real world data sets, Xiong and Chen (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, the $k$-nearest neighbors of a training data are computed first. Then the similarities of one sample from the 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 the following 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
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 on distinguishing additional classes. One advantage 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. Therefore, 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 effortlessly deals with the hugely multiclass nature of visual object recognition
3. From a theoretical point of view, it has the remarkable property wherein, under very mild conditions, the error rate of a kNN classifier tends toward the Bayes optimal as the sample size tends toward 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 rapidly decreases 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 kNN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their relevance. The main weakness of this technique is that performance is subject to the often ad hoc choice of the 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. However, due to the shortcomings faced by kNN, an imperative need is felt for a more robust tool with high efficiency and accuracy.

On the basis of the kNN approach, a novel tool VBClassif ver.1.0, for classification of epidemiological data of vector-borne diseases, was developed [Figure 1].

VBClassif is a novel efficient classification algorithm (designated as VBClassif 1.0), which classifies the records of those affected by Filariasis. This software has been utilized to classify up to a 1,00,000 records and the effective classification yield percentage is 94%.

METHODOLOGY

The data on various parameters like age, sex, class, and so on, of different individuals affected by Filariasis is

Figure 1: Snapshot showing VBClassif 1.0 application window
considered. The program is trained with a small portion of data designated as train data and is then tested for efficiency by executing on the test data, which is expectedly large. The algorithm is tested with minimal train data and observed to see if the efficiency increases or remains unaltered.

Dataset

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

Data normalization

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

Algorithm: The following is the main algorithm.

1. Start

2. From the entire given test data
   1. Separate ‘0’ class records and ‘1’ class records in the train data
   2. Take a record from the test data and calculate the Euclidean distance for each record in the train data for both the classes, separately
   3. Take least distance in ‘0’ class and ‘1’ class
   4. Compare the least distance in both the classes; put the query record in the class that is having the least distance
   5. If distances are equal those are taken as unclassified records
   6. Repeat steps 2 to 5 for every record in the test data.

3. Now, on the unclassified records from the above-mentioned stage perform the following steps
   7. Take the least distance record (say \( r_1 \)) in the ‘0’ class and least distance record (say \( r_2 \)) in the ‘1’ class, records that are classified.
   8. Take the record from the unclassified list that is having minimum distance (say \( u_1 \)).
   9. If \( u_1 \) is less than both \( r_1 \) and \( r_2 \) then take the minimum of \( r_1 \) and \( r_2 \).
   10. If \( u_1 \) is less than then assign \( u_1 \) to ‘0’ class. If \( u_1 \) is less than \( r_2 \) than assign \( u_1 \) to ‘1’ class.
   11. Repeat steps 3 and 4 for all unclassified records from step 2 of this method.

4. Again, on the unclassified records from the above-mentioned stage perform the following steps
   12. Calculate least \( L_1, L_2 \) and highest \( H_1, H_2 \) distances in ‘0’ and ‘1’ classes respectively.
   13. Calculate average \( r^* \) as \( r = \frac{H_1+L_2}{2} \).
   14. Calculate the distance (\( d_1 \) and \( d_2 \)) between a record in unclassified list and the record that is having \( H_1 \) distance and \( L_2 \) distance.
   15. If \( d_1 < r \) 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 the unclassified list and also for those in the true negative list.

5. On the unclassified list perform the following operations
   17. Calculate the distances between a record in the unclassified list and the remaining records in the same list.
   18. If \( x_1, x_2, x_3, \ldots, x_n \) are records in the unclassified list calculate the distances \( (x_1, x_2), (x_1, x_3), \ldots (x_1, x_n) \).
   19. Take the least distance (\( d_1 \)) from these calculated distances.
   20. Take least distances (\( r_1, r_2 \)) in the ‘0’ and ‘1’ classes respectively.
   21. Assign \( d_1 \) to the class to which \( m \) belongs.
   22. Repeat steps 1 to 5 for every record in the unclassified list and also in the true negatives list.

6. Again on the unclassified list perform the following steps
   23. Calculate the Euclidean distance between the origin and the record in the unclassified list.
   24. If \( x_1, x_2, x_3, \ldots, x_n \) are fields in the record then \( R_i = \sqrt{(x_1)^2 + (x_2)^2 + \ldots + (x_n)^2} \).
   25. Divide every field value by \( R_i \)
      \[ \frac{x_1}{R_1}, \frac{x_2}{R_1}, \frac{x_3}{R_1}, \ldots, \frac{x_n}{R_1} \]
   26. Calculate the Euclidean distance between a record with new field values in the unclassified list and every record in the classified records of ‘0’ and ‘1’ classes.
   27. Take least distance in the ‘0’ class and ‘1’ class.
   28. Compare the least distance in both classes and put the query record in the class that is having the least distance.
   29. Repeat the steps from 1 to 6 for every record in the unclassified list and the true negative list.

7. End of algorithm
RESULTS

The results obtained with different magnitudes of train data are shown in the Table 1:

The algorithm has performed effectively, compared to other well-known classification algorithms such as the K-nearest neighborhood (kNN) algorithm. The comparison results are tabulated in Table 2.

The results in Table 2 help one to conclude that it is not necessary to always have a large data of 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 classifications and predictions.

When applied to data sets with train data of different sizes, such as, 300, 500, 600, and 1100 records (with percentages being 6.7, 12, 13.5, and 25 of 4470), VBClassif ver 1.0 has rendered a classification accuracy of 98.4, 91.3, 92.77, and 83.38, respectively, with the percentage of classification being 100, while the well-known k-nearest neighborhood (kNN) algorithm could classify the above-mentioned test data achieving an accuracy of 96.82, 92.42, 87.26, and 81.72, with different sizes of train data as mentioned earlier.

The tool demonstrates a high percentage of classification accuracy as compared to the well-known kNN, which is a desired feature. VBClassif is a menu-driven and user-friendly, robust tool, which even a novice can apply, to analyze a 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 disease, and hence, provides an effective way of emulating the disease.

CONCLUSION

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

REFERENCES

1. Choi Serene Hyun-Jin, Nieminen Timo A, Bahr Mark, Bahr Nan. Improving Behaviour Classification Consistency: A Technique From Biological Taxonomy. In Australian Association for Research in Education Annual Conference, 2002.
2. Graham M., Kennedy JB, Hand C. The challenge of visualising multiple overlapping classification hierarchies 1999:42-51.
3. Wang D, Lv Y, Guo Z, Li X, Li Y, Zhu J, et al. Effects of replacing the unreliable cDNA microarray measurements on the disease classification based on gene expression profiles and functional modules. Bioinformatics 2006;22:2883-9.
4. Xiong H, Chen XW. Kernel-based distance metric learning for microarray data classification. BMC Bioinformatics 2006;7:2.
5. Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP: A structural classification of proteins database for the investigation of sequences and structures. J Mol Biol 1995;247:536-40.
6. Model F, Adorján P, Olek A, Piepenbrock C. Feature selection for DNA methylation based cancer classification. Bioinformatics 2001;17:157-64.
7. Rätsch G, Sonnenburg S, Schäfer C. Learning interpretable SVMs for biological sequence classification. BMC Bioinformatics 2006;7:9.
8. Orengo CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM. CATH—A hierarchical classification of protein domain structures. Structure 1997;5:1093-108.
9. Svétík V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP. Random forest: A classification and regression tool for compound classification and QSAR modeling. J Chem Inf Comput Sci 2003;43:1947-58.

Source of Support: Nil, Conflict of Interest: None declared.