Chapter 8
Handling Large Medical Data Sets for Disease Detection

Rahul Kala
Indian Institute of Information Technology and Management Gwalior, India

Anupam Shukla
Indian Institute of Information Technology and Management Gwalior, India

Ritu Tiwari
Indian Institute of Information Technology and Management Gwalior, India

ABSTRACT

The breakthrough in the field of intelligent systems has spread its fruits to the field of biomedical engineering as well; where a series of models are being applied to automatically detect diseases based on some parameters or inputs. The continuous research in this field has resulted in a large amount of database being created for many diseases which becomes very difficult to train. Also the number of attributes is under constant rise. This increases the dimensionality of the problem and ultimately leads to poor performance. In this chapter we deal with the methods to handle these situations. We discuss the mechanism to divide data between different sub-systems. We also discuss the method of division of the attributes to reduce the training time and complexity. The resultant systems are able to train better due to low computational cost and hence give better performance. We validated this with the Breast Cancer database from the UCI Machine Learning repository and found our algorithm optimal.

INTRODUCTION

Computational Intelligence has given rise to automation in numerous spheres. Bio-medical engineering is one of these spheres where the computationally intelligent systems have automated the task of detection of diseases. These systems are able to affectively detect the presence or absence of any disease. These systems take the various parameters as inputs that can potentially affect the decision regarding the presence or absence of disease. The system analyzes these inputs and then makes its decision. Every input or parameter affects the decision of the system to some extent. Some parameters are very important whereas the others may behave rather passive in nature. Ideally the larger the number of parameters, the more is the chance of detection of the disease. This is

DOI: 10.4018/978-1-61692-004-3.ch008

because of the fact that the system is given more information that may be consulted for decision making.

The detection system usually makes use of historic data to formulate rules regarding the analysis. The historic data is a collection of large amount of information regarding the presence or absence of disease in multiple scenarios. The system tries to extract valuable information or rules from this database by the process of machine learning. It then tries to generalize the extracted information or rules to the unknown data. The entire knowledge of the system to detect the disease is based on the learning of the historical data. Hence it can effectively only effectively predict the presence or absence of some disease if the behavior was recorded by the system sometimes earlier and presented at the time of learning as the historical database. In case the same does not hold, the system might not behave as desired and it may give any random or abrupt output. Hence ideally a larger historical database is preferred that can capture as much diversity as possible. The aim is that in the future any unknown input is related to one of the inputs in the historical database.

The large number of attributes in the system as well as the large number of training data size in the historical database seems to be a big boon for the system. However, this may have a negative impact on the system in terms of computational time. The larger number of attributes or data set instances has a multiplicative impact on the time of training. It may hence be possible that the system is unable to train itself in finite time. This would drop the efficiency of the system and the entire system would become sub-optimal. Further, the larger number of attributes makes the system very flexible. The entire system can now assume more behavior by different values of outputs for different inputs. In other words there is an increase in the number of parameters of the systems that require more data and time for training. There is always an associated risk of sub-optimal training as a result of the added attributes. Hence an increase in attributes or the number of inputs is not always desirable.

The larger number of instances of training data in the training data set also has negative affects apart from the time complexity being increased. More data added to the system may add unique characteristics to the system which is to be modeled. This may be an exception to the present state and behavior of the system. As a result the system would have to formulate new rules or learning to accommodate the input that acts as an exception. This would add to the complexity of the system. Further the complex systems are more computationally expensive and difficult to train due to the large system parameters. Also the training may be sub-optimal. This puts a serious limitation to the increase of data.

It may not always be the case that adding attributes or training instances is poor for the system. Some added attributes showing behavior of correlation to one or more of the already existing attributes may not make the system that sensitive. Similarly if all of the added inputs show normal behavior as the rest of the inputs in the database, there would not be much adverse affect on the system performance. In such cases we may consider the addition to play role in time only. It may at the same time help in optimal system tuning.

Modular Neural Networks (MNNs) (Shadabi, Sharma and Cox, 2006) are neural networks that exploit the modularity in the problem. The whole problem is divided into a set of modules which are Artificial Neural Networks (ANNs) in themselves. All the modules behave independent of each other. Each module tries to solve its part of the problem independently. After all the modules have solved the problem, the results are communicated to the integrator. The integrator is responsible for combining the results of the individual modules to form the result of the entire problem. In such a manner the problem is solved by the combined effect of the various modules. Each module is ultimately a neural network that has to undergo training before it can be put to use. The training of