Chapter 10

Conclusions and Future Work

The objectives of this thesis had been to improve the prediction accuracy and thus, the quality of the overall learning process. By quality we mean the improvement of criteria such as accuracy, complexity, robustness and interpretability. Many parts of our work such as discretization and topological learning can be applied in different types of classifiers e.g. naive-bayes classifier. However, our work focuses on decision tree based learning and thus, we introduce various preprocessing and transformation techniques such as discretization, fuzzy partitioning and dimensionality reduction to improve this type of learning. We divided our work into three parts and performed experiments to validate each claim made in each part.

10.1 Discretization Techniques by Resampling

Considering that the learning sample is just an approximation of the whole population, we argued that the optimal discretization built on a single sample set is not necessarily the global optimal one. Thus, we have used resampling such as bootstrap of build a distribution of discretization points obtained from B bootstrap samples. Thus, using these type of resampling approaches we developed three techniques.

The first technique based on building a resampled (B times) discretization point distribution has further two variations namely RDD and RSD. These two methods vary in the manner they extract selected discretization points from the built distribution. The former selects the most probable n points. This number n is calculated by considering the most

frequent 'number of intervals' obtained by the earlier repeated resampling. Whereas, the later smooths the distribution by using a moving average filter and then identifies 'peak regions' on most probable regions in the distribution. The most frequent points in each of these 'peak regions' are selected as discretization points.

The second method TBTD, works on top-down discretizations and applies a Bagging technique at each node (subsample) and selects the best point by a voting procedure. The original sample is split at this point and the sample procedure is recursively applied to the next two subsamples until a stopping criteria is met.

The third method builds probability density functions $f(X|C)$ of class distributions on an attribute X by resampling the learning sample. It further smooths the class distributions and identifies the decision boundaries as the final discretization points. This method is more suitable for naive-bayes classifiers because of its tendency to generate a larger number of intervals.

Detailed experiments were performed and showed that our resampling based approaches tend to give a better discretization estimate in terms of achieving better discretization quality. RDD, RSD and TBTD methods improved the discretization variance, robustness, prediction accuracies and provided smaller number of intervals. Thus, arrived nearer to a global optimal solution (FisherŠs optimal algorithm). In the case of DBD the number of intervals were much higher than the other three methods and thus, also exhibit larger variance. They worked better for naive-bayes classifiers than decision trees. Whereas the other methods worked better in the case of decision trees because they used MDLP discretization as the base discretizer, which is known to work better for decision trees than naive-bayes.

Among other methods except for Chi-Merge and Balancedgain, the other methods provide small variations in terms of prediction rates. MDLPC performs the best in terms of number of intervals and time complexity, MODL and Fusinter were not far behind. Fisher's method is the most expensive computationally while our DBD approach is the least. However, RSD and TBTD are more expensive than all the methods (except Fisher), but this phenomenon can be controlled by the number of bootstrap samples. We argue that the availability of high computational power these days can help us in focusing more on achieving better discretization quality.

10.2 Soft Discretization for Fuzzy Decision Trees

Two fuzzy decision tree techniques are presented that use fuzzy partitioning and a fuzzy entropy criterion for tree growing. The first technique uses resampling based fuzzy partitioning during the tree building process. This is an extension of the RSD discretization technique and uses the same motivation whose aim by using resampling is to build a discretization point distribution from which close to optimal discretization points can be obtained. We identify regions of high probability of finding discretization points which contribute in building fuzzy partitions during the tree building process.

In the later method, a top-down recursive fuzzy partitioning is presented. Here, we use fuzzy partitions only when necessary i.e. when the region around the cut point is vague and fuzzy. Otherwise, we argue that the split could become too fuzzy or we can search the whole attribute space for nothing. We justify the use of a heuristic to build fuzzy partitions by showing the high time complexity of optimal partitioning solutions.

We show by comparing with other classical and fuzzy decision tree methods that our fuzzy decision trees produced better accuracy with a greater ability to cater for large data noise and variance. In terms of classical decisions trees we used C4.5, CART and OC1 algorithms. Furthermore, we use C4.5 with continuous data and also with discretized values. In order to compare with other fuzzy partitioning techniques, we adopted three methods that all use the same fuzzy entropy criterion i.e. SDF-FDT, which forms binary fuzzy partitions by considering the standard deviation of the overall attribute values, BSF-FDT searches for the best width of β once the α is identified (based on fuzzy partitioning as in [21]) and optimal-FDT that uses a fuzzy adaptation of fisher's algorithm for fuzzy partitioning.

Our approaches RSF-FDT outperformed the other fuzzy methods including our DSF-FDT, in terms of accuracy, robustness and lower variance. However, in terms of model complexity the classical decision trees generated much simpler models. Thus, there is a trade-off of complexity vs accuracy. We also show that our solution is comparable to aggregation techniques such as Bagging and Boosting.

10.3 Classification by Manifold Learning

We derive a non-linear dimensionality reduction technique using manifold learning for the purpose of classification. We try to see whether using such techniques improve the overall decision tree learning. During the manifold process we use a technique based on spatial autocorrelation statistics to estimate the value of K and stress its importance in building the manifold structure. Then we apply ISOMAP and convert the original distances to geodesic distances on the graph. Next, we choose the resulting number of features and apply a decision tree based classification method for learning. For unseen examples either we do the entire process again or use the help of a neural network based extreme learning machine to convert the unseen data into labeled form, which improves in time complexity but suffers in classification accuracy.

On comparison with classical classification methods our technique performed better on classification accuracy, model complexity and comprehensibility due to dimension reduction but suffers from the time consummation due to additional computations. We argue that on more complex data sets such the swiss roll this performance is bound to increase further. We also observe that oblique decision trees tend to perform better with our technique than classical methods.

10.4 Extensions and Improvements

As future work, we can examine the potential of resampling as prior distributions in naive-bayes classifiers. We also plan to perform detailed experiments in terms of time complexity. One extension could be to compare with other fishers optimal algorithm having criterion other than Fusinter e.g. MODL criterion. Another extension could be the treatment of qualitative attributes. Currently, we are working on a solution in which we use create clusters of candidate discretization points from a discretization point distribution.

As future work we could add more datasets specially such datasets which are inherently imprecise and vague in order to exploit the true capacity of fuzzy decision tree learning. Experiments related to time complexity could be added specially in the case of multiple classes. We plan to take into account the comparison with fuzzy techniques using other

criteria as well e.g. [21, 17]. We also plan to work on fuzzy decision tree visualization.

Finally, in the case of SSM-Learn, there are many things that could be worked upon in future. We plan to add more extensive experiments on issues such as time complexity, model complexity and robustness. We could also add more classical data sets and furthermore special datasets e.g. complex forms of swiss roll dataset, that can really exploit the advantages of using manifold learning for classification. We could also compare more classifiers and similarly use this technique with other classifiers.

We are presently working on making an R package for fuzzy decision trees such that they can use multiple types of fuzzy partitioning e.g. user defined.