Summary 5-1 &5-2

Shaun Pritchard

Florida Atlantic University

CAP 6778

October -03-2021

M. Khoshgoftaar

Summary_5_1 - Classification Performance of Rank Aggregation Techniques for Ensemble Gene Selection

Nine different rank aggregation techniques were examined for selection of bioinformatics genes using hybrid ensemble learning. This research observed how to conduct ensemble feature selection processes that would perform multiple runs of a feature selection creating subsets of features which would then turn those results into a single feature subset that could be used for inference and classification. Using nine different rank aggregation techniques, research proves to find an optimum solution for building classification models to use gene microarray data.

Also, to note the data-sets where types of data used to distinguish between cancerous and non-cancerous cells between patients who did or did not respond well to specific cancer treatment. The techniques were tested using ensemble learning with 25 features techniques running 50 iterations each, with 11 bioinformatics data sets, and five classification learners.

Specifically, the objective was to determine whether the complexity of rank aggregation techniques contributed to better variance for classifying learners. The concept of rank aggregation simply means combining multiple results into one ranking. When a rank aggregation problem is applied to a set of candidates, many different order orderings are combined to obtain a better ranking.

To eliminate bias, the authors applied 5 k-fold cross validation under four runs of Hybrid Diversity Ensemble technique. Five learners were used: K-nearest neighbors (5-NN), Logistic regression (LR), Support Vector Machine (SVM), Multilayer Perceptron (MLP), and Nave Bayes (NB).

A total of twenty-five feature selection techniques were evaluated twice with fifty iterations, divided into three main groups: Threshold-based Feature Selection (TBFS) Techniques, First Order Statistics based feature selection (FOSBFS), and commonly-used techniques(CU).

- Within (TBFS) where Eleven of the techniques (Area under ROC curve, Deviance, F–Measure, Geometric Mean, Gini Index, Kolmogorov-Smirnov statistic, Mutual Information, Odds Ratio, Power, Probability Ratio, and Area Under the Precision-Recall Curve)
- Within (FOSBFS) Seven techniques used were Fisher Score, Fold Change Ratio, Fold Change Difference, Wilcoxon Rank Sum, Significance Analysis of Microarrays, Welch T Statistic, and Signal to Noise.
- Within (CU) used the Chi-Squared, Information Gain, Gain Ratio, ReliefF, ReliefF-W, Symmetric Uncertainty, and SVM–RFE techniques.

Rank aggregation techniques used were Mean, Median, Highest Rank, Lowest Rank, Stability Selection, Exponential Weighting, Enhanced Borda, Round Robin, and Robust Rank Aggregation.

Given the complexity of this experiment, I was surprised by the significance of the results. Findings showed that for seven of the techniques (Enhanced Borda, Exponential Weighting, Highest Rank, Mean, Median, Round Robin, and Stability Selection) in combination of learner and feature subset outperformed all the rest. In contrast, two of the performance techniques (Lowest Rank and Robust Rank) did not have a combination that outranked in performance. Results showed that the other rank aggregation techniques were statistically indistinguishable from each other.

Summary_5_2 - Stability and Classification Performance of Feature Selection Techniques

This study used four different imbalanced datasets, nine chosen features, and seven filter-based feature ranking techniques. The techniques used included; chi-square (CS), information gain (IG), gain ratio (GR), two forms of the ReliefF algorithm (RF and RFW), symmetrical uncertainty (SU), and signal-to-noise (S2N) were applied based on changes to the datasets implemented with a consistency index to determine stability in ranking while comparing rankings to classification performance for a contrast of significance.

It should be noted that consistency indexes utilize sequential forward selection(SFS). The SFS implements and builds a feature upon each subset repeatedly to determine their consistency. In addition to implementing these ranking techniques, the datasets consisting of defect prediction metrics were used to evaluate the significance of feature selection techniques for defect prediction. Feature selection is needed to select a subset of features that minimizes the prediction errors of classifiers for best results in testing the stability and model performance.

Software defect prediction focuses on determining which modules need extensive testing and which have defect risk. A software metric is thereby gathered during the development process, which contains information about a particular project. Inferences are then made about faulty software modules based on metrics collected. Studies have shown that these models perform better when irrelevant and redundant metrics (features) are removed from the data. Therefore, feature selection stability is important to this topic and the results of this experiment.

The experiment's main goals were to use these techniques on the given datasets to consider the stability of feature selection over the classification techniques by comparison of the stability and model performance and feature selection techniques used together on real-world software metrics data.

Final results revealed that the number of instances removed from the consistency index had a diverse effect on the stability of feature ranking techniques. According to the results, the fewer instances removed from a dataset, the less the selected features would have changed. Using the AUC performance metric, LR proved to be the most accurate classifier when evaluating classification against stability. The experimental results showed signal-to-noise(S2N) ranker was the best ranker in terms of model performance. The results showed that the ReliefF was the most stable feature selection technique but it performed significantly worse than other rankers.

Ultimately this research determined that filter ranking techniques do play an important role in feature selection implementation and stability of software defect metrics.