Summary 4-1 &4-2

Shaun Pritchard

Florida Atlantic University

CAP 6778

September -28-2021

M. Khoshgoftaar

Summary 4-1 -Feature Selection with High-Dimensional Imbalanced Data

Using five different bioinformatic microarray expression datasets of high dimensionality and class imbalance. Nine filter-based feature selection methods were tested for performance. Essentially, the research proved to experiment with six commonly used feature selection filters and three classifier Performance-based metrics used in feature selection. Also, the data implemented only three standard data sampling techniques for inference.

The research calculated the differences between the attribute rankings of each technique based on the ranking correlation between attributes. The ranking correlation makes it possible to determine which ranking is more important based on the deferential similarity resulting from the filter techniques. The goal of this study was to better understand the similarities and differences among feature selection methods. It is also important to note that this experiment focused on deriving value from the feature attribute rankings of the imbalanced datasets relative to the class attributes.

This research implemented the use of six feature selection methods to determine whether there is a statistically significant difference between the expected frequencies of filter-based rankings such as ($\chi 2$) chi-square, information gain (IG), gain ratio (GR), Symmetric Uncertainty, Relief-W, and ReliefF. Also, the research implemented three threshold base filters ROC Curves(ROC), F-Measure (F), and geometric mean (GM). The three Data Sampling Techniques implemented were composed of random sampling techniques such as oversampling (ROS), random undersampling (RUS), Synthetic Minority Oversampling Technique (SMOTE), and Wilson's editing (WE).

To determine the degree of similarity between the feature ranking techniques, and determine the Frobenius norm distance, which is the square root of the sum of squared distances. Kendall's Tau Rank Correlation was applied as the ranking correlation statistic. The primary objective of this work is to compare the performance of the various filter techniques.

The most formidable results show that the performance of the $\chi 2$, gain ratio, information gain, and symmetric uncertainty are all highly correlated on average to one another. as were the ReliefF and ReliefF-W which showed moderate correlation, and the three performance metric-based techniques AUC, GM, and F are also moderately correlated. Considering the final results, it was recommended further work should be conducted to investigate performance-metric-based filters. They produced only moderate correlation compared to very highly correlated six basic feature selection techniques which is significant.

Summary 4-2 -How Many Software Metrics Should be Selected for Defect Prediction?

This was empirical research, using experimentation, aimed at minimizing the number of feature attribute software metrics that should be used to build a software defect prediction (SDP) model for a given system using feature selection.

SDP is one of the most effective ways of reducing development costs and analyzing software quality. During the software development lifecycle, developers collect data about software defects. There is a strong correlation between the distance metric between samples and the performance of machine learning prediction models based on SDP. In addition, most samples are usually highly class imbalanced. Selecting software metrics systematically beforehand will likely improve performance in defect prediction models.

In many existing prediction models based on machine learning, the distance metric between samples has a significant impact on the performance of the SDP model for this reason the research is proposing to implement a threshold-based feature selection technique to remove irrelevant and redundant feature attributes of software metrics.

Based on data from a real-world SDP project, three commonly used classifiers are used in this experiment, and the results demonstrated that a defect predictor can be built using only three metrics, and that model performance increases with a significance of 98.5%.

The experiments were conducted on three groups of software data sets, each group having three separate releases. Utilizing five Threshold-Based filter-based Feature Selection techniques (TBFS) to select different sizes (subsets) which were based on five different performance metrics; Mutual Information (MI), Kolmogorov-Smirnov,(KS), Deviance (DV), Area Under the ROC (Receiver Operating Characteristic) Curve (AUC), and Area Under, the

Precision-Recall Curve (PRC). While being implemented with three different classifying learners Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), and Logistic Regression(LR). Which were measured with the classifier performance metrics of the Area Under ROC (Receiver Operating Characteristic) curve (AUC). The MLP learner showed that its best inference model was built with four features selected by the AUC ranker. The KNN learner showed that its best inference model was built with seven features selected by the AUC ranker. The LR learner showed that its best inference model was built with only three features selected by the AUC ranker. Overall, the best classification model was built with (LR) logistic regression implementing three features selected by the AUC ranker.

I find it personally interesting that the MLP, which has hidden layers, implements 3 nodes in which each node's activation function relating the inputs of each unit to its output is the same as for logistic regression. I thought while reading the research document that MLP would have surely won?

Nonetheless, the final results showed logistic regression to be the dominant learner while determining significance in the findings of the experiment that after removing 98.5% of the available number of software metrics (feature attributes) the defect prediction models performed way better than when all the features were used. This was significant and proved that fewer software metrics could be achieved with feature selection.