Ross Bollinger

Professor Nathalie Japkowicz

5 November 2018

CSC 480

Assignment #2 Documentation

**1. Introduction**:

In today’s world of advanced technology and computation, data is at a constant flow. As a result, it can be very difficult for big companies to keep up with the amount of data coming from various web sources. It is important to be able to know where specific data is coming from in the flow for things like bandwidth allocation and traffic shaping. This is where machine learning comes in. Using machine learning to classify data from different apps is a very effective process and one that has a large number of benefits. Most importantly, it provides the best way to accurately classify massive data sets that can result from the data flow of the internet. This paper has two main goals. The first is to classify active flow data based on the app it was generated from. This is to show that using machine learning is indeed a good method of active data flow classification. The second is to classify data based on whether it is from active or passive flows.

There are two data sets that will be used for the entirety of the experiment. The first is ActiveBiFlowData. This data set consists of 118,020 instances of data collected from an active flow from app users that chose to allow their data to be shared. It is classified by the several apps that each instance came from. The apps are as follows: QQ, WeChat, Facebook, Weibo, Youku, TencentVideo, MgTV, Browser, JdShop, VipShop, QQMail, YahooMail. It is a bi-feature data set. The second is PassiveBiFlowData. This data set consists of 9,617 instances of data collected from a passive flow from apps that were opened on a phone, but not being used. It is also a bi-feature data set. Both these data sets are, however, imbalanced in some way. This is because the data is not equally distributed among the twelve apps being used. This imbalance will need to be accounted for during experimentation.

To test the first goal, only the first data set will be used. It will be run through a variety of different machine learning classification algorithms with three different feature selection methods. This is to achieve the best possible feature selection/classification algorithm pair possible with this particular data. Finally, statistical tests will be performed to test the significance of the performance of each algorithm.

To test the second goal, both data sets will be used. The two data sets will be concatenated together with new classifiers of “passive” and “active”. This new data set will then be tested using the same algorithms used in the previous section, as well as the same feature selection methods. Statistical testing will be used on the performances of these algorithms as well.

The Orange Visual Tool for data mining will be used in all experimentation.

**2. Experimentation**

**2A. Classifying Active Data Based on Originating App**

In this section of experimentation, I sampled 40% of the data set due to how large it is, then ran it through several classification algorithms. This sampling method was also included to solve class imbalance since Orange automatically does this when sampling a percentage of data. Each time I did this, I used a different feature selection method to select the five most relevant features from the sample. The classification algorithms used are as follows:

* kNN (k Nearest Neighbor)
* Tree
* SVM (Support Vector Machine)
* Random Forest
* Neural Network
* Naïve Bayes
* AdaBoost (A boosting algorithm)

The feature selection methods used are as follows:

* Information Gain – based on the expected amount of information
* ANOVA – differencing the average values of features across classes
* FCBF (Fast Correlation Based Filter) – uses entropy and identifies redundancy due to pairwise correlations between features

I recorded the Area under the Receiver operating curve (AUC), Classification Accuracy (CA), the Classification Precision (P), Recall (R), the F1-Score (a weighted, harmonic mean between P and R) as the performance metrics. The F1-Score is especially important because it is useful when class imbalance exists.

**Table 1.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature Selection | Algorithm | **AUC** | **CA** | **P** | **R** | **F1-Score** |
| **Info Gain** | **kNN** | 0.826 | 0.570 | 0.567 | 0.570 | 0.567 |
| **Tree** | 0.763 | 0.536 | 0.538 | 0.536 | 0.534 |
| **SVM** | 0.459 | 0.103 | 0.122 | 0.103 | 0.100 |
| **Random Forest** | 0.857 | 0.579 | 0.579 | 0.579 | 0.575 |
| **Neural Network** | 0.839 | 0.507 | 0.519 | 0.507 | 0.478 |
| **Naïve Bayes** | 0.754 | 0.387 | 0.348 | 0.387 | 0.347 |
| **AdaBoost** | 0.846 | 0.562 | 0.560 | 0.562 | 0.560 |
| **ANOVA** | **kNN** | 0.825 | 0.562 | 0.561 | 0.562 | 0.561 |
| **Tree** | 0.784 | 0.558 | 0.561 | 0.558 | 0.556 |
| **SVM** | 0.472 | 0.118 | 0.164 | 0.118 | 0.126 |
| **Random Forest** | 0.846 | 0.578 | 0.577 | 0.578 | 0.575 |
| **Neural Network** | 0.839 | 0.521 | 0.541 | 0.521 | 0.492 |
| **Naïve Bayes** | 0.765 | 0.402 | 0.360 | 0.402 | 0.338 |
| **AdaBoost** | 0.804 | 0.554 | 0.554 | 0.554 | 0.554 |
| **FCBF** | **kNN** | 0.822 | 0.542 | 0.537 | 0.542 | 0.539 |
| **Tree** | 0.765 | 0.537 | 0.539 | 0.537 | 0.535 |
| **SVM** | 0.419 | 0.105 | 0.150 | 0.105 | 0.099 |
| **Random Forest** | 0.856 | 0.577 | 0.571 | 0.577 | 0.572 |
| **Neural Network** | 0.847 | 0.522 | 0.554 | 0.522 | 0.490 |
| **Naïve Bayes** | 0.795 | 0.468 | 0.418 | 0.468 | 0.425 |
| **AdaBoost** | 0.804 | 0.540 | 0.538 | 0.540 | 0.539 |

As seen by Table 1, the best feature selection method varies with each algorithm:

Information Gain:  
On average, the kNN, Random Forest, and AdaBoost algorithms had the best results compared to results with other feature selection methods.

ANOVA:  
On average, the Tree and SVM algorithms had the best results compared to results with other feature selection methods.

FCBF:  
On average, the Neural Network and Naïve Bayes algorithms had the best results compared to results with other feature selection methods.

These results show that Information Gain was the best feature selection method used in this experiment. However, the Random Forest classification algorithm showed to have outperformed every other algorithm in every way and with every feature selection method. I performed the sign test on the results of the Random Forest algorithm against the results of every other algorithm. This was done to the results from each feature selection method. In all three methods, the test showed that all the results from the Random Forest algorithm were statistically significant, with an average z-score of 2.24 at a 95% confidence interval.

**2B. Classifying Active from Passive Data**

For this section of experimentation, I concatenated the ActiveBiFlowData and PassiveBiFlowData sets. To do this, I simply removed any class features from each data set and replaced them with a “Data Type” class that contained either “Active” or “Passive” respectively as class instances. Due to how large the Active data set was compared to the passive data set, I only sampled the exact number of instances there are in the passive data set from the active one. This means that I sampled 9,617 instances from the active set to match the 9,617 instances in the passive set. This also solves class imbalance because now I have a perfect 50% split of the data between the two classes.

I repeated the experimentation method from the previous section. This includes the same feature selection methods.

**Table 2.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature Selection | Algorithm | **AUC** | **CA** | **P** | **R** | **F1-Score** |
| **Info Gain** | **kNN** | 0.948 | 0.893 | 0.893 | 0.893 | 0.893 |
| **Tree** | 0.864 | 0.881 | 0.881 | 0.881 | 0.881 |
| **SVM** | 0.484 | 0.553 | 0.553 | 0.553 | 0.553 |
| **Random Forest** | 0.962 | 0.899 | 0.899 | 0.899 | 0.899 |
| **Neural Network** | 0.866 | 0.792 | 0.792 | 0.792 | 0.792 |
| **Naïve Bayes** | 0.766 | 0.665 | 0.670 | 0.665 | 0.663 |
| **AdaBoost** | 0.884 | 0.884 | 0.884 | 0.884 | 0.884 |
| **ANOVA** | **kNN** | 0.881 | 0.806 | 0.806 | 0.806 | 0.806 |
| **Tree** | 0.812 | 0.827 | 0.828 | 0.827 | 0.827 |
| **SVM** | 0.478 | 0.560 | 0.561 | 0.560 | 0.559 |
| **Random Forest** | 0.937 | 0.862 | 0.862 | 0.862 | 0.862 |
| **Neural Network** | 0.805 | 0.731 | 0.732 | 0.731 | 0.731 |
| **Naïve Bayes** | 0.712 | 0.652 | 0.658 | 0.652 | 0.649 |
| **AdaBoost** | 0.830 | 0.830 | 0.830 | 0.830 | 0.830 |
| **FCBF** | **kNN** | 0.948 | 0.893 | 0.893 | 0.893 | 0.893 |
| **Tree** | 0.864 | 0.881 | 0.881 | 0.881 | 0.881 |
| **SVM** | 0.497 | 0.553 | 0.553 | 0.553 | 0.553 |
| **Random Forest** | 0.964 | 0.900 | 0.900 | 0.900 | 0.900 |
| **Neural Network** | 0.866 | 0.792 | 0.792 | 0.792 | 0.792 |
| **Naïve Bayes** | 0.766 | 0.665 | 0.670 | 0.665 | 0.663 |
| **AdaBoost** | 0.884 | 0.884 | 0.884 | 0.884 | 0.884 |

As seen in Table 2, the best feature selection method again varies with each algorithm. This time, however, there were some ties between the Information Gain and FCBF methods for the best results from each algorithm. The ties occurred with the kNN, Tree, Neural Network, Naïve Bayes, and AdaBoost algorithms. The only algorithms that didn’t follow this trend were the SVM and Random Forest algorithms, which both had their best results with the FCBF method. This shows that the FCBF is the better feature selection method this time around, which is probably due to there being only two classes to work with this time around.

Yet again, the Random Forest algorithm managed to outperform every other classification algorithm in every single case. The sign test was performed on each set of results from each feature selection method. In every case, the results of Random Forest, with an average z-score of 2.24, were significant at a 95% confidence interval.

**3. Conclusion**  
 The results of all experimentation done on this data show that the Random Forest classification algorithm is the most ideal machine learning method when dealing with data from both passive and active data flows. When it came to feature selection, Information Gain was the best method for classifying active data based on the apps it was recorded from, and FCBF was the best method for classifying active from passive data. The Random Forest model has been known to cause overfitting, so it is possible that it only was able to do this well on the training data, and not any new data.

There could have been many mistakes made through experimentation. The size of the sample in experiment 1 could have contributed to its results because there was not enough training data for the other algorithms to work with. Maybe the feature selection methods chosen for both experiments were not the best possible methods to test. These possible errors create potential next steps for this research, and the number of possibilities is not surprising.