Data parsing

* Data hole filling
* Normalization
* Nominalization

Machine Learning Algorithms

* naiveBayes
* logistic
* libLINEAR
* libSVM
* ensembleSelection
* logitBoost
* supersampling

The raw data was a combination of nominal and numeric classes. The first step was to fill in missing data fields. In nominal classes, a special null value was assigned. In numeric classes, the average was assigned. Next the nominal classes and the numeric classes were separated because they were preprocessed differently. The numeric classes were normalized over the range 0 to 1 because literature says that this increases the performance of many classification algorithms. The nominal data of each class was put into a dictionary and then enumerated in order to make it readable by MATLAB classifiers. The nominal values were then turned into strings so that they could be processed by Weka.

Weka made it easy to try many different learning algorithms quickly. Due to the nature of our data, we were very interested in comparing the performance of traditional classification algorithms with meta-classifiers such as boosting and ensemble learning. The first learning algorithms we tried were logistic regression, libLINEAR, and naïve Bayes. Talk about logistic regression. We found that changing the C parameter for libLINEAR had little to no effect on the classification performance. Standard libLinear had rather poor performance, with an AUC of \_\_\_\_. Standard naïve bayes performed quite well with an AUC of .737.

Next we tried some meta-classifiers, specifically adaBoostM1, logitBoost, and ensemble selection. The motivation for using meta-classifiers is that they can adaBoostM1 and logitBoost performed similarly, resulting in similar AUC scores of .729 and.746 respectively. Because of this we decided to stick with logitBoost as our boosting algorithm of choice. Ensemble selection was first tried with ten REPTree classifiers with unique parameters. REPTree is a very simple decision tree learner, making it an ideal first choice for evaluating a lower bound on classification performance. It only had an AUC of .696, however it had great precision. The next ensemble selection was tried with J48, a decision tree based on maximizing information gain. It had an AUC of .71. Next, we tried ensemble selection with naïve bayes, since naïve bayes performed well as a classifier on its own. Unfortunately it didn’t even perform as well as original naïve bayes, with an AUC of .722. It did however have great recall. Last, we tried ensemble selection with a mix of naïve bayes, J48, and REPTree classifiers. This resulted in the highest AUC yet of .752. Additionally, it had good precision.

To improve the performance of our classifiers we tried balancing our data so that there were equal positive and negative examples. Literature has shown that balancing data can greatly improve classifier performance. In order to do this the positive examples were super-sampled. Discuss results.