Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

Enhancing Ensemble Learners with Data Sampling on High-Dimensional Imbalanced Tweet Sentiment Data

Joseph D. Prusa, Taghi M. Khoshgoftaar, Naeem Seliya

jprusa@fau.edu, khoshgof@fau.edu, nseliya@gmail.com

Abstract

High dimensionality and class imbalance are two important concerns when training tweet sentiment classifiers. Feature selection techniques reduce dimensionality by selecting an optimal subset of features. Class imbalance can be addressed by either using classifiers that are robust to the impact of class imbalance, such as those trained with an ensemble learning technique, or by using data sampling techniques to create a sampled training set with a more balanced class ratio. These separate techniques can be combined together to address both class imbalance and high-dimensionality; however, it is unclear if it is necessary to use data sampling and ensemble techniques together as both are used to target class imbalance. In our study, we investigate if the addition of random undersampling to Select-Boost (feature selection and boosting) significantly improves the performance of sentiment classifiers trained on imbalanced tweet data. We evaluate classifiers trained using four base learners and three feature subset sizes across two highly dimensional imbalanced datasets. Our results show, for tweet sentiment, the inclusion of random undersampling significantly improves classification performance and indicates this may be more noticeable on datasets with greater levels of class imbalance.

Introduction

Tweet sentiment data, being real world data, is frequently class imbalanced. Class imbalance, where there are majority and minority classes, can negatively impact classification performance as classifiers may be biased towards classifying new, unseen instances as belonging to the majority class. Another challenge, inherent to tweet sentiment classification, is high dimensionality, which refers to datasets with a very large number of available predictive features. Since features for tweet sentiment datasets are extracted from user-posted tweets, large numbers of textual features are frequently generated. Such datasets can potentially have tens of thousands of features (Saif, He, and Alani 2012). Classifiers trained on highly dimensional datasets are prone to over fitting (Prusa, Khoshgoftaar, and Dittman May 2015), resulting in poor classification performance. Additionally, many of the features may not be beneficial to use and removal of such features decreases the computational costs associated with classifier building and prediction. Together, class imbalance and high dimensionality present two key data science challenges found in tweet sentiment data, and these typically have a negative impact on classifier performance if not addressed.

We can alleviate the impact of both of these issues using specific machine learning techniques. Feature selection techniques intelligently select a subset of features, reducing data dimensionality and potentially improving classifier performance (Prusa, Khoshgoftaar, and Dittman May 2015). Multiple data mining approaches exist that improve classifier performance on imbalanced data. Data sampling can be used to sample instances from the imbalanced dataset to create a sampled dataset with a more balanced distribution of majority and minority classes. This minimizes or eliminates majority class bias as classifiers will be trained from the newly created and more balanced dataset (Van Hulse, Khoshgoftaar, and Napolitano 2007). An alternative approach is to use ensemble learning algorithms. Ensemble learners train multiple classifiers through data or algorithmic diversity and combine them into a single classifier that performs better and is more robust to data quality issues, such as noise or class imbalance, than its constituent classifiers (Prusa, Khoshgoftaar, and Dittman 2015).

Using combinations of the above techniques, we can address both data concerns. Feature selection in combination with ensemble learners can be used on any dataset; however, data sampling is only necessary on imbalanced data. Additionally, both data sampling and ensemble learners may likely improve classification performance on imbalanced data. Moreover, we determined in a preliminary investigation, training classifiers with boosting (a popular ensemble learning algorithm) results in a greater increase in classification performance than data sampling when considered alone. For these reasons, we choose to compare feature selection in combination with an ensemble learner against the combination of all three techniques to determine if it is necessary to include data sampling or if ensemble algorithms are sufficient to address class imbalance. We did not consider the combination of feature selection with data sampling, since our preliminary study indicated combining feature selection with ensemble learning yielded better results.

In this paper, we train classifiers using four base learners with both Select-Boost (feature selection and boosting) and Select-RUS-Boost (feature selection, Random Under-Sampling and boosting). We compare the performance of these two approaches on two tweet sentiment datasets. Both datasets were constructed by sampling 9000 instances from the sentiment140 corpus (Go, Bhayani, and Huang 2009) to have a specified class ratio. The first is sampled to have a 20:80 positive to negative sentiment class ratio, while the second has a 5:95 class ratio, to respectively represent data with moderate and severe levels of class imbalance. Our results indicate that using RUS in addition to feature selection and boosting improves performance for the majority of the learner and feature subset combinations. We conducted statistical tests verifying our experimental results and determined the observed performance improvement is significant. To the best of our knowledge, this is the first study to combine feature selection, data sampling and ensemble techniques to target the key data issues of class imbalance and high dimensionality present in tweet sentiment data, and the first to evaluate if the addition of RUS improves performance of ensemble learners on high-dimensional, imbalanced sentiment data.

The remainder of this paper is organized as follows. The Related Works section provides a background on the machine learning techniques employed in this study, how they have been used in the context of tweet sentiment, and existing literature the benefit of their combination. In the Methodology section, we explain how our datasets were created, our chosen feature selection technique, boosting approach, data sampling, and how we combine these techniques. Additionally, we explain how we train and evaluate our classifiers. The Results section presents and discusses the performance of our classifiers and statistical tests to verify our observations. Finally, notable observations from our study, as well as suggestions for future work, are presented in the Conclusions section.

Related Works

Feature selection techniques are used to reduce data dimensionality by choosing a subset of features, and have been shown to be useful when training classifiers from tweet sentiment data. Using the sentiment140 corpus, Saif et al. (Saif, He, and Alani 2012) used information gain to select subsets with between 42 and 34,855 features, and used Naïve Bayes with each subset size to train sentiment classifiers. They determined there was no advantage to using more than 500 features with their dataset, greatly reducing the computational resources needed to train classifiers on their dataset. In another study, Chamlertwat et al. (Chamlertwat et al. 2012) found using information gain to perform feature selection could improve tweet sentiment classifiers trained with Support Vector Machines. Additionally, word frequency is commonly used when extracting features as a crude form of preliminary feature selection and is present in many studies on tweet sentiment where n-grams are used as features (Go, Bhayani, and Huang 2009).

Boosting is a popular ensemble learning algorithm that has been demonstrated to improve performance for tweet sentiment classifiers. Silva et al. (Silva, Hruschka, and Hruschka Jr 2014) compared the performance of Support Vector Machine (SVM) and Multinomial Naïve Bayes (MNB) with and without AdaBoost. They found MNB with boosting performed better than either classifier without boosting. While they demonstrated boosting improves classification performance, they did not use ensemble techniques in the context of class imbalance.

Class imbalance has been found to negatively impact tweet sentiment classifier performance (Hassan, Abbasi, and Zeng 2013), (Silva, Hruschka, and Hruschka Jr 2014); however, data sampling can be used to improve the performance of sentiment classifiers on imbalanced data. One of the first experiments involving data sampling and sentiment classification (product reviews, not tweets) was conducted by Li et al. (Li et al. 2011). Their results showed that using RUS offered superior performance compared to using the full training data, and also found RUS outperformed random oversampling.

These techniques can be combined together to improve classifier performance on datasets suffering from both high dimensionality and class imbalance. Khoshgoftaar et al. (Khoshgoftaar et al. 2013) proposed a hybrid machine learning technique that combines feature selection with RUS and boosting to create Select-RUS-Boost. They found this technique significantly improves classifier performance on high dimensional imbalanced bioinformatics datasets compared to RUS-Boost (Random UnderSampling and boosting). Unlike tweet sentiment data, bioinformatics datasets frequently contain very few instances; however, they share the potential issues of high dimensionality and class imbalance, thus this hybrid technique is of interest in the domain of tweet sentiment classification.

In this study we extend previous investigations of using machine learning techniques to improve sentiment classifiers by combining multiple techniques together to investigate the interaction of feature selection, boosting and data sampling on tweet sentiment data. Unlike previous studies on class imbalanced sentiment data, we compare the combination of feature selection and boosting against the combination of feature selection, boosting and data sampling and evaluate if including data sampling provides a significant improvement in classifier performance when used in addition to feature selection on tweet sentiment data. This is of interest as ensemble learners are more robust with respect to being trained on imbalanced data than individual learners, thus it is of benefit to the data analyst to learn if it is necessary to combine data sampling and boosting.

Methodology

The following subsections describe our experimental design and methodology including: how our dataset was constructed, our implementation of feature selection in combination with boosting, feature selection with boosting and data sampling, the machine learning algorithms we use to train classifiers, our training methodology, and classifier performance evaluation metric.

Table 1: Datasets created from sentiment140 Twitter corpus

Class Ratio	# Pos	# Neg	# Features
20:80	1800	7200	5059
5:95	450	8550	5048

Datasets

We constructed two imbalanced datasets using instances from the sentiment140 corpus (Go, Bhayani, and Huang 2009). This corpus contains 1.6 million tweets that were automatically collected and labeled by searching for tweets containing emoticons (representations of facial expressions created with letters, numbers, symbols and punctuation marks). Tweets were given a sentiment label matching the polarity (either positive or negative) of the emoticon used as a query to find the tweet. This method of labeling can label large numbers of tweets automatically, but can result in noisy class labels as not all tweets will match the sentiment of the emoticon used to retrieve the tweet. While manually labeled datasets exist and are relatively less noisy, they contain a limited number of instances and cannot be used to create highly imbalanced datasets with a sufficient number of instances.

We constructed two datasets with different class ratios. Each contains 9000 instances sampled from the corpus to produce a dataset with a specified class ratio. We used unigram (individual word) features after cleaning the text of URLs, punctuation marks, symbols, excessive character repetitions and capitalization in an effort to standardize the text of the tweets. Features were extracted independently for each dataset. The first dataset has a 20:80 class ratio (1800 positive, 7200 negative) and has 5059 features. The second dataset was constructed by sampling 450 positive and 8550 negative instances to create a dataset with 9000 instances and a 5:95 positive:negative class ratio. Details for both datasets can be found in Table 1.

Boosting

Boosting is an ensemble learning technique that combines multiple learners in an effort to improve classifier performance. Boosting creates an ensemble of classifiers iteratively. In each iteration, a classifier is trained and applied to the training data. Misclassified instances are given a higher weight in the next iteration, so that the next classifier will be better at correctly classifying them. By making use of data or algorithm diversity, ensemble learning techniques can be used to train classifiers that are more robust to noisy or imbalanced data, resulting in higher predictive performance than their constituent learners (Prusa, Khoshgoftaar, and Dittman 2015).

In our study, we use the popular boosting algorithm AdaBoost.M1, implemented in $WEKA$ (Witten and Frank 2011). The default behavior of AdaBoost is to use re-weighting of instances to inform the algorithm of which instances are more important in each iteration. Re-weighting is not compatible with some base learners. So, in our version of AdaBoost, we use data sampling with replacement as an alternative to re-weighting. We create a new training dataset where instances are sampled in such a way that the distribution of instances matches the weights assigned to each instance in the previous boosting iteration. This allows us to use boosting with any base learner.

Data Sampling

We use Random UnderSampling (RUS) as it has been shown to be the best data sampling technique for many domains, including sentiment classification (Li et al. 2011). RUS randomly selects majority class instances and removes them from the dataset until the desired class distribution is achieved (Van Hulse, Khoshgoftaar, and Napolitano 2007), thus creating a new, more balanced dataset that is smaller than the original dataset. While removing instances reduces the information that can be used to train a classifier, for large datasets (such as those encountered when training tweet sentiment classifiers) this loss of instances should not be problematic and is outweighed by the benefit of training classifiers on a more balanced dataset. Oversampling techniques are less desirable when training classifiers on large datasets as they increase the size of the dataset. Additionally, RUS was found to perform better than random oversampling with tweet sentiment data (Li et al. 2011). In this study, we elected to use RUS with a 50:50 post sampling class ratio which was selected based on our prior work (Prusa et al. 2015).

Feature Selection

Feature selection techniques seek to select an optimal subset of features, reducing data dimensionality and potentially improving classifier performance. They can be either filter based (feature rankers and subset selection filters), wrapper based, embedded, or hybrid. Filter-based feature subset evaluation, wrapper based techniques and hybrid techniques require significantly more computational resources than filter-based rankers. Thus, in this study, we use filter-based rankers which are well suited for tweet sentiment classification, as they are relatively fast and scalable (Prusa, Khoshgoftaar, and Dittman May 2015). These feature selection techniques rank features and then select a subset consisting of the top ranked features.

In this study, all experiments were conducted using the area under the Receiver Operating Characteristic (ROC) curve to perform feature selection. This technique is a threshold based feature selection technique that uses the trade-off between true positive rate and false positive rate (when considering a single feature and a simple classification rule) to rank features. This technique was selected as preliminary experiments showed it to have the highest performance on imbalanced data with feature subset sizes of 100, 150 and 200 features.

Select-Boost and Select-RUS-Boost

In this study, we implement two combinations of the above feature selection, boosting and data sampling techniques i.e., Select-Boost (S-Boost) and Select-RUS-Boost (S-RUS-Boost). S-Boost combines feature selection with boosting.

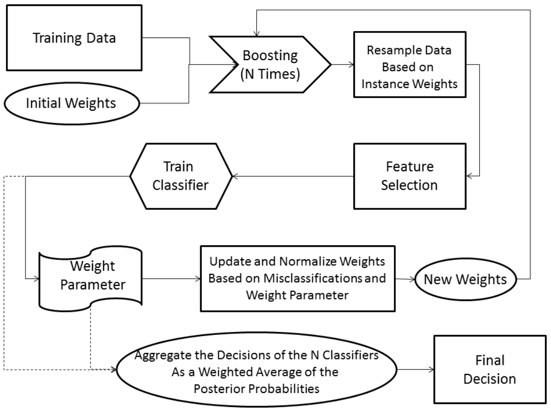
Training Data
Initial Weights
Boosting (N Times)
Resample Data Based on Instance Weights
Train Classifier
Feature Selection
Weight Parameter
Update and Normalize Weights Based on Misclassifications and Weight Parameter
New Weights
Aggregate the Decisions of the N Classifiers As a Weighted Average of the Posterior Probabilities
Final Decision

Figure 1: Select Boosting

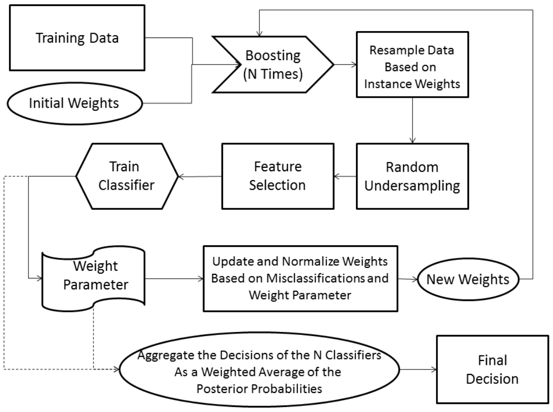
Training Data
Initial Weights
Boosting (N Times)
Resample Data Based on Instance Weights
Random Undersampling
Feature Selection
Train Classifier
Weight Parameter
Update and Normalize Weights Based on Misclassifications and Weight Parameter
New Weights
Aggregate the Decisions of the N Classifiers As a Weighted Average of the Posterior Probabilities
Final Decision

Figure 2: Select RUS-Boost

This allows us to train classifiers that benefit from being trained in an ensemble while also addressing problems associated with high-dimensionality such as over fitting. S-Boost performs feature selection in every iteration of the boosting algorithm after data has been resampled based on instance weights. Figure 1 provides an outline of our algorithm.

S-RUS-Boost is similar to S-Boost, but performs random undersampling before feature selection. An overview of the algorithm is presented in Figure 2. In this study, S-RUS-Boost and S-Boost used 10 iterations, selected based on preliminary experiments which found using more iterations did not significantly improve classification performance. When performing S-RUS-Boost data was resampled to a 50:50 post sampling class ratio. Considering other post-sampling class ratio values is out of scope for this paper, largely due to paper size limitations.

Classifiers

We elected to use four machine learning algorithms: C4.5 decision tree, Naïve Bayes (NB), Support Vector Machine (SVM) and Logistic Regression (LR) as base learners. All algorithms were implemented in $WEKA$ (Witten and Frank 2011). As these four classifiers are commonly used we will not explain each in detail due to space constraints; however, parameter changes made to default settings are as follows.

When creating our decision trees we set "$nopruning$" and "$LaplaceSmoothing$" to "$true$" as this has been found to improve the performance of C4.5 (Van Hulse, Khoshgoftaar, and Napolitano 2007). SVM was trained using a linear kernel and the complexity constant within $WEKA$'s implementation of SVM was set to 5.0. Additionally, "$buildLogisticModels$" was set to "$true$", enabling proper probability estimates to be obtained (Witten and Frank 2011). Naïve Bayes and Logistic Regression were trained using default parameters. All changes to default parameters were found to improve classifier performance in preliminary experiments.

Cross-Validation and Performance Metric

We employ 5-fold Cross-validation (CV) when training classifiers, which randomly splits the data into five equal partitions and uses four folds as training data, while the remaining fold serves as a test dataset. This is repeated until each fold has been used for validation. The whole process is repeated four times to reduce any bias due to how the dataset was split when creating partitions. We evaluate the classification performance of each fold using Area Under the Receiver Operating Characteristic Curve (AUC). This is not to be confused with the feature selection technique employed in this paper, denoted as ROC. AUC is a measure of performance across all possible error cost ratios and class distributions and provides an effective numeric representation of how well a classifier will perform on imbalanced data (Witten and Frank 2011).

Results and Analysis

Using the above methodology and datasets, we compared the performance of our selected learners using S-Boost and S-RUS-Boost. The results of our experiment are presented in Table 2 and Table 3, subdivided by feature selection subset size. Average AUC scores are displayed for each combination of learner and technique. For each subset size and learner, the model with the highest AUC is indicated in **boldface**.

From Table 2 it can be seen that using S-RUS-Boost yields higher classification performance than S-Boost for three learners with either 200 or 150 features. However, the converse is true when using 100 features. While C4.5N still performs better with S-RUS-Boost than S-Boost for 100 features, the remaining learners perform better with S-Boost. The highest performance observed for each learner, excluding SVM, and the highest performance of any of the 24 classifiers uses S-RUS-Boost; however, relatively little difference is observed between the two approaches on the 20:80 imbalanced data. C4.5N shows the greatest improvement when using S-RUS-Boost compared to S-Boost, as its AUC increases by over 1% for each subset size. The remaining learners show little variation between S-Boost and S-RUS-Boost. In particular, SVM has no difference greater than 0.3%, and when trained using a subset of 200 features the difference between approaches is less than 0.01%. Using S-RUS-Boost performs better for 7 out of the 12 possible com-

Table 2: Classification Performance on 20:80 Data

Learner	200		150		100	
	S-Boost	S-RUS-Boost	S-Boost	S-RUS-Boost	S-Boost	S-RUS-Boost
C4.5N	0.753872	**0.765038**	0.749626	**0.761263**	0.744255	**0.75379**
NB	0.789331	**0.789874**	0.784638	**0.785598**	**0.77958**	0.776984
SVM	**0.787535**	0.787503	0.783897	**0.783492**	**0.776798**	0.773783
LR	0.786425	**0.786938**	**0.784868**	0.782651	**0.779494**	0.77647

Table 3: Classification Performance on 5:95 Data

Learner	200		150		100	
	S-Boost	S-RUS-Boost	S-Boost	S-RUS-Boost	S-Boost	S-RUS-Boost
C4.5N	0.700992	**0.719709**	0.699501	**0.720146**	0.699406	**0.712003**
NB	0.730024	**0.747793**	0.726179	**0.745159**	0.725788	**0.73806**
SVM	0.735874	**0.736832**	0.731983	**0.733453**	**0.732633**	0.729942
LR	**0.73019**	0.719427	**0.729354**	0.72953	0.726956	**0.734158**

Table 4: ANOVA results for 20:80

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Tech.	0.00054	1	0.00054	5.1	0.0243
Learner	0.07427	3	0.02476	234.93	0
Subset	0.01055	2	0.00528	50.07	0
Error	0.04984	473	0.00011		
Total	0.1352	479			

Table 5: ANOVA results for 5:95

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
Tech.	0.00789	1	0.00789	13.43	0.0003
Learner	0.05423	3	0.01808	30.75	0
Subset	0.00065	2	0.00032	0.55	0
Error	0.27807	473	0.00059		
Total	0.34085	479			

binations of learner and subset size, but these differences are small (excluding C4.5N). Classifiers uniformly perform better with more features.

Results for the highly imbalanced 5:95 class ratio dataset are presented in Table 3. It can be observed that S-RUS-Boost performs better than S-Boost for more of the possible learner and feature subset combinations than was observed on the 20:80 imbalanced dataset. Additionally, the performance difference between the two approaches is greater on this dataset. C4.5N with S-RUS-Boost achieves AUCs over 2% higher than with S-Boost, and NB performs 1 to 1.5% higher with S-RUS-Boost than S-Boost. LR performs 2% higher using S-Boost rather than S-RUS-Boost with 200 features, but its highest observed performance results from using S-RUS-Boost and 100 features. Again, SVM shows little change in AUC between the two approaches. Unlike the 20:80 imbalanced dataset, classifiers trained on the second dataset do not always yield the highest AUC when using 200 features. The highest AUC for NB is observed with a subset size of 150 and (as previously mentioned) LR achieves its highest AUC with 100 features.

For both levels of imbalance, S-RUS-Boost has higher performance than S-Boost for the majority of learner and feature subset size combinations, with the difference being far more noticeable for the 5:95 imbalanced dataset. However, the benefit of using S-RUS-Boost over S-Boost depended greatly on choice of base learner. C4.5N showed the clearest improvement, SVM shows little difference between the two technique, and LR was observed to frequently perform better when using S-Boost, though its highest performance on each dataset was observed when using S-RUS-Boost. Comparing the differences in performance between S-Boost and S-RUS-Boost for each learner on both levels of imbalance it appears that it is more important to include RUS on datasets with high levels of class imbalance as the performance difference between approaches is larger. It is possible that training a robust classifier using boosting may be sufficient for datasets with low levels of class imbalance, but data sampling is needed for higher levels of imbalance.

ANOVA and Tukey's HSD

We conducted a three-factor ANalysis Of VAriance (ANOVA) with a 5% confidence interval to determine if the choice of S-Boost or S-RUS-Boost techniques significantly impacts performance, and also considered learner and feature subset size as additional ANOVA factors. Results for our ANOVA on the 20:80 imbalanced dataset are presented in Table 4 and show that all three factors are significant. A second ANOVA test was conducted for the 5:95 imbalanced dataset, and results are presented in Table 5. Again, all three factors are significant. The ANOVA tests also indicate that the choice of classifier has a significant impact on performance (this is not surprising). Additionally, the differences in AUC observed between subset sizes of 200, 150 and 100 are significant. We also conducted Tukey's Honestly Significant Difference (HSD) tests, shown in Figures 3 and 4. Both show, averaging across all learners and subset sizes, classifiers trained with S-RUS-Boost perform significantly better than those trained with S-Boost. Thus it is beneficial to include RUS in addition to boosting and feature selection when training sentiment classifiers from highly dimensional and imbalanced tweet data.

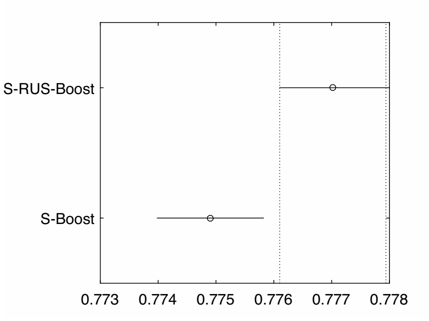

Figure 3: HSD Test for techniques on 20:80 Dataset

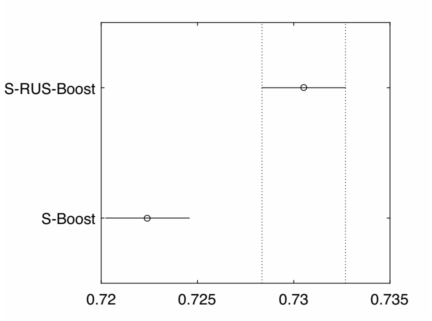

Figure 4: HSD Test for techniques on 5:95 Dataset

Conclusions

Class imbalance and high dimensionality are two important concerns when training a classifier from tweet sentiment data. While these issues can degrade classification performance, machine learning techniques exist addressing each problem. Feature selection reduces data dimensionality. Data sampling and ensemble learners both improve classifiers trained on imbalanced data. These techniques can be combined to address both issues, and boosting can be used alongside with RUS if a single technique is insufficient at addressing the impact of class imbalance.

In this study, we compare the performance of using feature selection in combination with boosting (S-Boost) and feature selection with RUS and boosting (S-RUS-Boost). From our experiment, we observed that including RUS significantly improves classifier performance and should be included in addition to boosting when training sentiment classifiers from imbalance tweet data. Additionally, the benefit of RUS appears to be more noticeable on datasets with greater levels of class imbalance.

Future work should investigate additional levels of class imbalance. It would be useful to know at what level of imbalance RUS is effective, since its removal of instances could potentially be detrimental to classifier performance (especially if boosting sufficiently addressed the level of imbalance), and how dataset size impacts the benefit of RUS. Additionally, this study should be extended to include additional datasets to see if results generalize.

Acknowledgment: We acknowledge partial support by the NSF (CNS-1427536). Opinions, findings, conclusions, or recommendations in this material are the authors and do not reflect the views of the NSF.

References

Chamlertwat, W.; Bhattarakosol, P.; Rungkasiri, T.; and Haruechaiyasak, C. 2012. Discovering consumer insight from twitter via sentiment analysis. *J. UCS* 18(8):973–992.

Go, A.; Bhayani, R.; and Huang, L. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford* 1–12.

Hassan, A.; Abbasi, A.; and Zeng, D. 2013. Twitter sentiment analysis: A bootstrap ensemble framework. In *Social Computing (SocialCom), 2013 International Conference on*, 357–364. IEEE.

Khoshgoftaar, T. M.; Dittman, D. J.; Wald, R.; and Napolitano, A. 2013. Contrasting undersampled boosting with internal and external feature selection for patient response datasets. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 2, 404–410.

Li, S.; Wang, Z.; Zhou, G.; and Lee, S. Y. M. 2011. Semi-supervised learning for imbalanced sentiment classification. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, 1826.

Prusa, J.; Khoshgoftaar, T. M.; Dittman, D. J.; and Napolitano, A. 2015. Using random undersampling to alleviate class imbalance on tweet sentiment data. In *Information Reuse and Integration (IRI), 2015 IEEE International Conference on*, 197–202. IEEE.

Prusa, J.; Khoshgoftaar, T. M.; and Dittman, D. J. 2015. Using ensemble learners to improve classifier performance on tweet sentiment data. In *Information Reuse and Integration, 2015 IEEE International Conference on*, 252–257. IEEE.

Prusa, J. D.; Khoshgoftaar, T. M.; and Dittman, D. J. May 2015. Impact of feature selection techniques for tweet sentiment classification. In *Proceedings of the 28th International FLAIRS conference*, 299–304.

Saif, H.; He, Y.; and Alani, H. 2012. Alleviating data sparsity for twitter sentiment analysis. CEUR Workshop Proceedings (CEUR-WS. org).

Silva, N. F.; Hruschka, E. R.; and Hruschka Jr, E. R. 2014. Biocom usp: Tweet sentiment analysis with adaptive boosting ensemble. *SemEval 2014* 123.

Van Hulse, J.; Khoshgoftaar, T. M.; and Napolitano, A. 2007. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, 935–942. New York, NY, USA: ACM.

Witten, I. H., and Frank, E. 2011. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition.