2016 15th IEEE International Conference on Machine Learning and Applications

An Investigation of Ensemble Techniques for Detection of Spam Reviews

Brian Heredia*, Taghi M. Khoshgoftaar†, Joseph Prusa‡ and Michael Crawford§
Department of Computer and Electrical
Engineering and Computer Science
Florida Atlantic University
Boca Raton, Florida
Email: *bheredia@fau.edu, †khoshgof@fau.edu, ‡jprusa@fau.edu, §michaelcrawf2014@fau.edu

***Abstract*—Whether purchasing a product or searching for a new doctor, consumers often turn to online reviews for recommendations. Determining whether reviews are truthful is imperative to the consumer, as to not get misled by false recommendations. Unfortunately, it is often difficult, or impossible, for humans to ascertain the validity of a review through reading the text; however, studies have shown machine learning methods perform well for detecting untruthful reviews. Previously, no studies have examined the effects of ensemble learners on the detection of untruthful reviews, despite these techniques being effective in related text classification domains. We seek to inform other researchers of the effects of ensemble techniques on the detection of spam reviews. To this aim, we evaluate four classifiers and three ensemble techniques using those four classifiers as base learners. We compare the results of Multinomial Naïve Bayes, C4.5, Logistic Regression, Support Vector Machine, Random Forest with 100, 250, and 500 trees, and Boosting and Bagging using the base learners. We found that none of the ensemble techniques tested were able to significantly improve review spam detection over standard Multinomial Naïve Bayes and thus, are not worth the computational expense they inflict.**

***Keywords*—spam review, ensemble, random forest, boosting, bagging**

I. Introduction

New experiences can be daunting without preparation. In the past, consumers relied on publications or services when deciding on something new. Currently, the most common way of preparing for something new, whether it be a new restaurant, a new physician, a new location, or a plethora of new experiences, is via online reviews and recommendations. Yelp, TripAdvisor, and Amazon are a few of the many websites containing reviews and recommendations on specific topics. These reviews and recommendations are crucial to the performance and growth of businesses. Unfortunately, one prevalent issue found on these sites is the presence of untruthful (spam) reviews.

Spam reviews do not accurately represent the topic and are meant to mislead the reader. Spam reviews can be used to harm, or bolster, the reputation of a business or product. In 2014, the Canadian government estimated a third of all reviews are spam reviews [10]. The prevalence of spam reviews has led to a drop in the credibility of online reviews. Determining if a review is truthful, from a human perspective, is often difficult, if not impossible. Human readers do not see the relationship between words used in spam reviews and words used in truthful reviews. Due to this difficulty, methods for detecting spam reviews have become a point of interest in the last decade.

Most of the current methods for detecting spam reviews use a supervised learning approach [6]. Supervised learning consists of training a classifier on labeled data sets to create a model that identifies reviews as spam or not. These labeled data sets contain features describing each review and whether the review is spam. The most common features describing a review include the words found in the review text, the meta data found (rating, date, time, etc.), and reviewer-oriented features, which describe the user that created the review. Other features, such as Part Of Speech (POS), Linguistic Inquiry and Word Count (LIWC), and stylometric features, have also been used to identify untruthful reviews [17] [16] [22]. However, our research focuses on using solely text based features to determine whether a review is truthful. Current research has focused on data set creation, feature creation, and identifying the best classifiers; however, no study has looked at the usefulness of ensemble methods for classification of spam reviews.

Supervised learning methods normally use a machine learning algorithm to classify instances. However, a family of techniques, called ensemble learning, combines multiple classifiers to create a more robust and generalized classifier. Moreover, ensemble techniques have been shown to increase classifier performance [7] [8]. In previous works, it has been found that ensemble techniques, such as Bagging and Boosting, increase classification performance in the text classification domain [18] [20] [19]. Unfortunately, research on the usefulness of ensemble classifiers when detecting spam reviews has not yet been considered. This study will help inform other researchers of the effects of ensemble techniques in the domain of spam review detection. To the best of our knowledge, this is the first study that examines the effects of ensemble learning techniques in the domain of spam review detection.

This study provides an evaluation of several ensemble learning techniques on spam review detection using a comprehensive data set, which spans three domains: restaurants, hotels, and doctors. We present the results for three ensemble methods: Boosting, Bagging, and Random Forest (RF). Boosting and Bagging are conducted using the Multinomial Naïve Bayes (MNB), C4.5, Logistic Regression (LR), and Support Vector Machine (SVM) base learners. Random Forest is conducted using three forest sizes: 100 (RF100), 250

978-1-5090-6167-9/16 $31.00 © 2016 IEEE
DOI 10.1109/ICMLA.2016.38

(RF250), and 500 (RF500) trees. All models are built using four runs of five-fold cross-validation. Area Under the receiver operator characteristic Curve (AUC) is our chosen metric for measuring the performance of our models. Plain classification using the base learners are used as a baseline for comparison to ensemble techniques. Our results show MNB, with or without an ensemble framework, significantly outperforms all approaches using C4.5. Moreover, no statistical difference is observed between plain MNB, Boosting or Bagging using MNB, RF250, RF500, or Bagging using SVM and LR. However, we do observe that RF500 produces the highest AUC score. We observe a significant increase in performance when using ensemble techniques with LR, C4.5, and SVM, however, MNB does not benefit to the same degree as the other learners. We recommend using only the MNB base learner, without ensemble techniques, as MNB is faster, less computationally expensive, while offering similar performance when compared to ensemble methods.

The remainder of the paper is organized as follows. Section II contains works related to our topic of spam detection and ensemble learning. Section III summarizes our methodology, including data set information, ensemble techniques, base learners, cross-validation, and the performance metric. Section IV presents our results along with statistical analyses. Section V presents our conclusions, recommendations, and possible avenues for future work.

II. Related Work

Current efforts in spam review detection are focused on data set creation, feature creation, and determining the best classifiers. Ott et al. [17] contributed a data set created using Amazon Mechanical Turk (AMT) [4]. AMT is a service that provides consumers with an on-demand, scalable work force. AMT users were asked to create positive untruthful reviews for a hotel. This service was leveraged to create a data set consisting of 800 positive reviews. Half of these reviews were false, created using AMT. The remaining truthful half was obtained via TripAdvisor. Performance of Naïve Bayes and Support Vector Machine (SVM) classifiers were evaluated using POS, LIWC, unigram, bigram, and trigram features. SVM, using both LIWC and bigram features, had the highest performance, resulting in an accuracy of 89.8%. Unfortunately, the data set put forth in this study has been shown to poorly represent real-world situations [16]. Murkherjee et al. [16] found AMT created reviews are more easily identified because of the word distributions. Words used by AMT workers do not match the words found in real world reviews, making them easier to spot.

Li et al. [14] expanded upon the data set created by Ott et al. [17]. AMT was again leveraged to create reviews for doctors and restaurants. These untruthful reviews were added to the original data set, expanding the domains covered to hotels, restaurants and doctors. However, Li et al. [14] also obtained untruthful reviews from employees in each domain. Since these employees have domain knowledge, their reviews were more akin to real world reviews as they used key domain words. A classifier modeled off the Sparse Additive Generic Model (SAGE) was used for classification. The model was trained using the instances belonging to the hotel domain and tested on the instances belonging to the doctor domain. Results show accuracies of 64.7% and 63.4% when using LIWC and POS features, respectively. These results show classification, using this data set, to be a more difficult task, representing real world scenarios. To the best of our knowledge, this is the only publically available data set representative of real world spam reviews.

Shojaee et al. [22] employed stylometric features to identify untruthful reviews. These features consist of lexical and syntactic features. Lexical features focus on word or character use, while syntactic features represent the style of the writer, such as their use of the words “the”, “of”, “a”, etc. The authors used the data set put forth by Ott et al. [17]. Three different feature sets were created: one using solely lexical features, one with solely syntactic features, and a final set combining both. For each feature set, two classifiers were trained: SVM and Naïve Bayes. The best F-measure, 0.84, was obtained using the feature set containing the combination of lexical and syntactic features. Moreover, SVM outperformed Naïve Bayes in all three experiments.

Ensemble techniques have yet to make their way into the realm of spam review detection. However, ensemble techniques have been used in Prusa et al. [18] to improve sentiment detection in tweets. Tweet sentiment detection uses the text found in tweets to determine the sentiment the tweet portrays. Normally tweets are cleaned by removing punctuation and symbols. Tweets are similar to reviews as both are user generated, posted online, and somewhat informal. However, tweets are shorter and less directed than online reviews. Tweet sentiment data set features contain the text from the actual tweets, creating a large feature space, similar to spam review data sets. Traditional ensemble techniques, Boosting and Bagging, were applied within this study using seven base learners and two tweet sentiment data sets. One data set consisted of a large number of instances labeled automatically, while the other was a smaller subset of this larger data set labeled manually. On both data sets, use of an ensemble technique improved performance for the majority of base learners. On the larger data set, Bagging showed the largest increase in performance across all classifiers, except k-Nearest Neighbor. For the smaller data set, Bagging showed the highest increase in performance for four of the seven base learners, while Boosting had the largest increase over the remaining three. Overall, ensemble methods showed an increase in performance over base classifiers.

To the best of our knowledge, no other study has examined the usefulness and effectiveness of ensemble techniques in spam review detection. Our intention is to evaluate ensemble techniques for spam review detection and to summarize the benefits for other researchers. Our study is unique in that it examines the usefulness of Boosting and Bagging, using four base learners, and Random Forest, with three forest sizes, for the detection of untruthful reviews in this domain.

III. Methodology

A. Data Set

In our study, we use the data set found in Li et al. [14]. It has been shown that learners trained on completely synthetic fake reviews have been found less effective in practice [16], thus, having a data set that resembles a real world environment is crucial. The data set was created using AMT [4], with the

addition of expert reviews from employees within the domains. The addition of these expert reviews makes this data set unique and the only publicly available data set with a similar word distribution to real world reviews [14]. The data set consists of reviews for doctors, hotels, and restaurants. Details on the number of instances and distribution of spam and non-spam instances can be found in Table I. From Table I, we see the class distribution is mostly balanced, with slightly more spam instances. Since this data set was an expansion of the data set put forth by Ott et al. [17], the majority of the reviews come from the hotel domain.

Domain	Truthful	Spam	Total
Doctors	200	356	556
Hotels	800	1080	1880
Restaurants	200	200	400
Total	1200	1636	2836

TABLE I: Data Set Characteristics

Two examples of reviews from the data set can be found below. From a human standpoint, determining which is spam between the two from reading the text alone is near impossible. The following review is an untruthful spam review:

> *The rates at The Talbott Hotel were cheaper than I had expected, and that was my reason for booking a room. I had been prepared for service and a room similar to what I had experienced in the past, and I was quite pleased when I did stay. The room was neat and clean, and the halls were quiet at night. The traffic noise was muffled to the point where it was no problem sleeping either. I did ask one question at the service desk and they answered it nicely, which is good because normally hotel workers can be a bit snippy, especially at night. Overall I had no problems with The Talbott Hotel and I would stay at this here again if I were in the area a second time.*

The following review is a truthful review, created by a guest who stayed in the hotel:

> *My husband & I stayed at the Fitzpatrick in early June 2004 for my birthday-great hotel! Location provides easy walking to Navy Pier, Marshall Field, Michigan Avenue & John Hancock. Room seemed spacious even though its only about 300 sq ft. Lots of room in bathroom; comfortable bed; very quiet, upscale hotel. We would definitely stay here again!*

The data set contains instances, such as the reviews above, the text in the review, the sentiment associated with the review, the domain the review belongs to, and the class label. For our purposes, the sentiment and domain information were removed, as we are only interested in the effects of text on spam detection. To the best of our knowledge, this data set is the only available text based spam review data set representative of real world data. The final feature space consists of the text of the reviews, represented using a bag-of-words approach. To create this bag-of-words feature set, the StringToWordVector filter in Weka is used [11]. While StringToWordVector can output a variety of word vector representations, we elect to use a bag-of-words representation over TF-IDF as we found the former to perform better during preliminary studies. We do not consider more advanced methods of feature engineering as the focus of this paper is on machine learning techniques. The StringToWordVector filter in Weka returns the number of words specified in the WordsToKeep parameter based on word frequency. If a value larger than the number of unique words is specified, all unique words are used to create the model. We specify 25,000 words to encompass all the text in the reviews.

B. Ensemble Techniques

Three ensemble techniques are employed in this study: Boosting, Bagging, and Random Forest. These techniques are similar in that they combine multiple instances of a base learner to generate a more robust and generalized classifier; however, the process by which this is achieved is different.

AdaBoost is a popular boosting technique that has been shown to increase base learner performance. We elect to use this boosting algorithm in our experiments. Boosting uses an iterative approach [9], which first assigns weights to instances within the data set, then trains a model using a base learner. Once a model has been trained, the instances are reassigned weights, with higher weights being placed on misclassified instances. The subsequent model is trained using the new weights and, again, reassigns weights based on misclassified instances. This process repeats a predefined amount of times. In our study, we set the iterative process to repeat ten times, as preliminary experiments have shown no significant increase in performance past ten iterations. Finally, the results of the classifications are aggregated, using majority voting. Due to the re-weighting step, Boosting is not compatible with certain base learners; however, a separate approach, where the instances are re-sampled according to the weights, allows these base learners to be compatible with Boosting. This approach samples data, with replacement, in such a way that the probability of selecting an instance corresponds to its assigned weight.

Bagging, also known as bootstrap aggregating, creates bootstrap data sets using sampling with replacement (duplicate instances can be chosen) from the original data set. The bootstrap data sets are the same size as the original data set. In our study, ten bootstrap data sets were created from the original since preliminary experiments show no significant increase in performance when using more bootstraps. Classifiers are then individually trained on each of the bootstrap data sets and their results are aggregated using majority voting. Bagging has been shown to increase performance of weak classifiers, but adversely affect the performance of stable learners [3]. In previous text classification experiments, it has been shown that bagging significantly improves performance over a number of classifiers [18].

Random Forest is an ensemble technique which utilizes multiple unpruned C4.5 decision trees to classify instances [2]. Random Forest also creates bootstrap data sets using sampling with replacement, however, those data sets are each trained on a C4.5 decision tree. At each node within a tree, the C4.5 algorithm chooses the feature which best discriminates between the classes using information entropy. Information entropy measures the amount of information gained for a class when a feature is considered. Using this method, C4.5 chooses the features with the highest information gain towards the top

of the tree. Random Forest differs from Boosting and Bagging in that it also provides random feature subspace selection. At each node within a decision tree, a subset of features are considered for the decision split. Once all bootstrap data sets have been used for training a tree, the results are aggregated using majority voting and a final classification is achieved.

C. Base Learners

In our study, four base learners are used, both within ensemble techniques and as classifiers. These learners are implemented using the Weka toolkit [11].

Multinomial Naïve Bayes [15] falls under the category of Bayesian learners and is a derivation of the Naïve Bayes learner. Similar to the Naïve Bayes learner, Multionomial Naïve Bayes also uses the naïve assumption of feature independence. MNB assumes each feature is independent, however, in general, this is usually not the case, especially in text domains. The main difference between Naïve Bayes (NB) and MNB is the way in which the probabilities are calculated. In Multinomial Naïve Bayes for text classification, the instance (a document in this example) is assigned to the class which has the highest conditional probability of $P(C|X)$. To calculate this probability, a count is done of the words which overlap between the document and the class, and then the count is divided by the total number of words. If $P(C_1|X) > P(C_2|X)$ then the document is classified as C1, otherwise it is classified as C2. MNB has been shown to be effective at detection of spam online reviews [5].

C4.5 [21] creates a decision tree based on the features that discriminate the most between classes using information entropy. The information a feature provides in determining the class is measured as a decrease in entropy. The features which decrease entropy the most, thus maximizing information, are found towards the top of the tree. The leaves of the tree contain the final classification. C4.5 is implemented using "Laplace smoothing" and "no pruning" as this has been shown to increase performance in previous studies [23].

Support Vector Machine [12] attempts to find a hyperplane that divides the instances into two groups. The data may be transformed via a kernel function into linearly separable spaces, but this is not always the case. The transformations allow for nonlinear boundaries to be formed around the data. The best such hyperplane would be the one that maximizes the distance between the hyperplane and members of each class. For our models, the complexity constant "c" was set to 5.0 and the "buildLogisticModels" parameter set to "true."

Logistic regression [13] attempts to find a probabilistic relationship between the features and the class label, similar to linear regression. This process is different from linear regression as it is used for the task of classification. Logistic regression aims to create a probability function that uses features as inputs and returns the probability of that instance belonging to a class as an output. Logistic regression was chosen due to its simplicity and effectiveness. The formula for logistic regression can be found below:

$$\log \frac{P}{1-P} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n$$

where $\frac{P}{1-P}$ is the odds ratio, X_i is the value for that feature, and β_i is the coefficient associated with feature X_i.

D. Cross-Validation and Performance Metric

All models are trained using four runs of five-fold cross-validation. Cross-validation is chosen over a random sampling approach because it allows for the full data set to be used, both for training and testing. In five-fold cross-validation, the data set is split into five folds. Four folds are used to train the model, while the remaining fold is used as a test. This process is repeated five times, alternating which fold is used for testing. To combat bias, due to a chance split in the folds, we run four runs of five-fold cross-validation and average the results.

We elect to use the Area Under the receive operator characteristic Curve (AUC) as a performance metric [23]. This metric is chosen because it plots the tradeoff between true positive rate and false positive rate of the model across all decision thresholds. The larger the area under the curve, the better the performance of the model, with a perfect model having an AUC of 1 and a random model having an AUC of 0.5.

IV. Results

In this section, we present the results of our experiments and determine the effects of ensemble techniques on spam review detection. Table II presents the results of our experiments. The table shows the average AUC across four runs of five-fold cross-validation, grouped by learner and ensemble method. Table II-A presents results for None, which represents using only the base learner and no ensemble technique, Bagging, and Boosting. Table II-B presents the results for Random Forest using three different tree sizes. We see RF500 as the top performer, with an AUC of 0.907. We observe that MNB, either with or without ensemble techniques, produces a higher AUC than the approaches using any other base learner. To determine whether this difference in performance is significant, these results are tested for statistical significance, at the 95% confidence level, using ANalysis Of VAriance (ANOVA) and a Tukey's Honestly Significance Difference (HSD) test [1].

Ensemble	MNB	C4.5	SVM	LR
None	0.900	0.729	0.867	0.857
Boosting	0.902	0.827	0.879	0.876
Bagging	0.900	0.823	0.889	0.888

(A) Average AUC for Boosting and Bagging

	100 Trees	250 Trees	500 Trees
Random Forest	0.876	0.899	0.907

(B) Average AUC for Random Forest

TABLE II: Average AUC for All Models

We first examine the performance of Boosting and Bagging, and compare their performance to the base learners. Then, we determine statistical significance between Random Forest models. Finally, we compare the best Random Forest models with the Boosting and Bagging approaches.

A. Bagging and Boosting

To determine whether Boosting or Bagging statistically improve over plain classification, we perform a two-factor ANOVA and a Tukey's HSD test. A Tukey's HSD test is used to conduct pairwise similarity tests to determine if the differences between two given learners is statistically significant. Table III presents both the ANOVA and Tukey's HSD results for Boosting, Bagging and plain classification using the base learners. The two-factor ANOVA test is presented in Table III-A. The factors for this ANOVA are ensemble techniques and base learners. Ensemble techniques include Boosting, Bagging, and None, while the base learners are MNB, SVM, C4.5, and LR. The ANOVA results show there is a significant difference between base learners, ensemble methods, and the interaction between them.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Base Learners	3	0.40	0.13	561.61	0.0000
Ensemble	2	0.07	0.03	137.63	0.0000
Base Learners:Ensemble	6	0.07	0.01	51.07	0.0000
Residuals	218	0.05	0.00		

(A) ANOVA Results for Boosting, Bagging and Plain Classification

	Classifier	Group	AUC	stdev
1	Boosting MNB	a	0.902	0.009
2	Bagging MNB	a	0.900	0.014
3	MNB	a	0.900	0.013
4	Bagging SVM	ab	0.889	0.011
5	Bagging LR	ab	0.888	0.011
6	Boosting SVM	bc	0.879	0.012
7	Boosting LR	bcd	0.876	0.015
8	SVM	cd	0.867	0.014
9	LR	d	0.857	0.027
10	Boosting C4.5	e	0.827	0.013
11	Bagging C4.5	e	0.823	0.019
12	C4.5	f	0.728	0.018

(B) Tukey's HSD Results for Boosting, Bagging, and Plain Classification

TABLE III: ANOVA and Tukey's HSD for Boosting, Bagging and Plain Classification

From Table III-B, we see the average AUC and Tukey's groups across all experiments for Boosting, Bagging, and plain classification. There are eight distinct groupings: 'a' through 'f'. If the classifiers share the same letter, then the difference in performance is not statistically significant. Boosting, Bagging, and plain classification using MNB fall into group 'a', while Bagging using LR or SVM fall into group 'ab'. This indicates that Bagging with LR or SVM shows no significant difference with a classifier in group 'a' or group 'b'. Boosting using SVM falls into group 'bc' and Boosting using LR falls into group 'bcd'. SVM and LR with no ensemble techniques fall into groups 'cd' and 'd', respectively. Finally, we see Boosting and Bagging using C4.5 in group 'e' and C4.5 in group 'f'.

From the results we observe that ensemble techniques significantly increase performance of SVM, LR, and C4.5. Boosting does not significantly increase performance when using SVM, however, Bagging significantly increases performance over the base SVM learner. Bagging has a significantly higher AUC than the base LR model, while Boosting is not significantly different than the base LR classifier. Both Boosting and Bagging significantly increase performance over C4.5. As ensemble techniques combine multiple learners, the result is a more robust and stable model.

We see that the best base learner is MNB. Moreover, the differences between MNB with and without ensemble techniques are not significant; however, not using an ensemble technique is faster and less computationally costly. Boosting with MNB does produce the highest AUC and the lowest standard deviation. Bagging does not increase performance of MNB but does increase standard deviation. Ensemble techniques do increase performance over C4.5, LR, and SVM when detecting spam reviews. However, ensemble techniques do not increase performance over MNB.

B. Random Forest

This section presents the results for the three Random Forest sizes. To determine what number of trees is best for RF, a one-factor ANOVA is performed and results are presented in Table IV-A. The ANOVA test shows that the difference in the number of trees used within RF is significant. To determine what number of trees is best, a Tukey's HSD is conducted and results are presented in Table IV-B. Table IV-B shows two statistically different groups, RF500 and RF250 fall into group 'a' and RF100 falls into group 'b'. Thus, RF500 and RF250 statistically surpass RF100, but the difference between RF500 and RF250 is not significant. However, we note that RF500 does produce a higher AUC but also has a runtime nearly twice as long. We elect to use RF500 and RF250 for further comparison with Boosting, Bagging, and plain classification.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Trees	2	0.01	0.01	19.92	0.0000
Residuals	57	0.01	0.00		

(A) ANOVA Results for RF100, RF250, and RF500

	Trees	Group	AUC	stdev
1	500	a	0.907	0.016
2	250	a	0.899	0.017
3	100	b	0.876	0.015

(B) Tukey's HSD Results for RF100, RF250, and RF500

TABLE IV: ANOVA and Tukey's HSD for Random Forest

C. Comparisons

In this section, we determine the overall best method for predicting spam reviews by examining the performance of the best models from our experiments in previous sections. We elect to measure the performance of the two best Random Forest models (RF500 and RF250) against the top performing Boosting and Bagging models (plain MNB, Boosting using MNB, and Bagging using MNB). These classifiers are selected as they have been shown to be the best performers in our experiments, but have not yet been compared. Statistical significance of the results is determined using a two-factor ANOVA test, where the two factors are learner (RF500, RF250, MNB) and ensemble type (RF, Boosting, and Bagging). The results of the ANOVA test are presented in Table V. The

ANOVA results show there is no significant difference between choice of learner or ensemble technique. These results indicate that there is no statistical difference when choosing learner or ensemble technique to use. Table VI shows the AUC values associated with each of the models mentioned. We see that RF500 has the highest AUC, however, this difference is not significant from any of the models listed in the table.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Learner	2	0.00	0.00	2.40	0.0964
Ensemble Type	2	0.00	0.00	0.11	0.8962
Residuals	95	0.02	0.00		

TABLE V: ANOVA Results for top performers

	Classifier	AUC	stdev
1	RF500	0.907	0.016
2	Boosting MNB	0.902	0.009
3	MNB	0.900	0.013
4	Bagging MNB	0.900	0.014
5	RF250	0.899	0.017

TABLE VI: AUC and Standard Deviation results for top performers

Table VII presents training times for SVM, MNB, and RF. We are comparing these learners as they are used as a base learner in at least one classifier belonging to group A in previous Tukey HSD tests. We include training times for SVM and MNB using bagging, boosting, and no ensemble technique (plain classification), as well as results for RF250 and RF500. Each classifier was trained for 1 run of 5-fold cross validation on a laptop with a 2.20GHz i7-4702HQ CPU and 8.00 GB of RAM running Weka exclusively. Minimum, maximum, mean and standard deviation are presented for the training times of the five folds for each classifier.

We observe that MNB is the fastest training algorithm, taking an average of 1/100th of a second to train each fold. SVM requires 12.05 seconds per fold. Thus, as a base learner MNB is close to 1,000 times faster to train than SVM for our data set. Using bagging with MNB requires approximately 10 times as long to train as plain classification, using boosting requires close to 15 times as long. For SVM, bagging takes approximately 7.5 time longer and boosting takes 11.5 times as long. RF250 and SVM with boosting have very similar training times. As expected RF500 takes approximately twice as long to train as RF250, since the runtime of random forest should scale linearly with number of trees. Due to its fast training time and high performance when used without an ensemble technique, we recommend using MNB when training a classifier for this domain.

Ensemble techniques create robust classifiers, however, they do nothing to combat the issue of a high dimensional feature space found in text classification. Spam review detection suffers from high dimensionality, since the features used to determine if a review is spam are composed of the words found in the review. Boosting and Bagging do not address the problem of high dimensionality, however, RF does have random feature subspace selection at every node within a tree. Random feature subspace selection allows RF to choose random features to examine at every node within a decision tree. As RF and Bagging take similar approaches, with the creation of bootstrap data sets and aggregating classifiers to form a final decision, we can compare the performance of Bagging using C4.5 and RF. It is likely random feature subspace selection is one of the reasons we observe the highest AUC when using Random Forest.

TABLE VII: Training Time for Different Learners and Ensemble Techniques

Learner	Ensemble	Min	Max	Mean	S.Dev
MNB	plain	0.009	0.011	0.01	0.001
	bag	0.094	0.109	0.103	0.009
	boost	0.141	1.56	0.147	0.009
SVM	plain	11.484	12.469	12.05	0.367
	bag	85.734	91.313	88.228	2.553
	boost	131.891	143.234	137.584	4.275
RF	250	133.516	140.531	137.619	2.558
	500	262.719	268.625	265.478	2.644

V. Conclusion

As the popularity of online reviews and recommendations continues to grow, so do the number of untruthful reviews. In recent years, the credibility of online reviews has dropped. A method for detection of untruthful (spam) reviews becomes increasingly necessary as the number of spam reviews continues to rise. Current research in the field of spam review detection focuses on the use of supervised learning techniques to classify a review as spam. Moreover, current research has been focused on determining the best classifier, data set creation, and feature creation. Ensemble techniques, which combine learners to form a more robust and generalized classifier, have yet to be examined within the realm of spam review detection.

In this study, we consider three different ensemble methods for the detection of untruthful reviews: Boosting, Bagging and Random Forest. Boosting is performed using the MNB, SVM, LR and C4.5 base learners and ten iterations. Bagging is also performed using the MNB, SVM, LR, and C4.5 learners with ten bootstrap data sets. The performance of three Random Forest sizes are examined and the best performing sizes are used for comparison against the best performing Boosting, Bagging, and plain classification methods. Our results show there is a significant increase in performance for ensemble techniques when using SVM, LR, and C4.5. However, MNB with no ensemble technique performs significantly better than SVM, LR, C4.5 with Boosting, and C4.5 with Bagging. Our results show there is no statistical difference between Bagging, Boosting and using no ensemble technique with MNB; however, we do find that MNB is less computationally costly and faster than implementing ensemble techniques. RF500 is observed to have the highest AUC, but is not significantly better than using MNB without ensemble techniques and is more computationally costly. While ensemble techniques have been found to be beneficial in other text classification domains, we observed no significant improvement compared to MNB with no ensemble technique. Thus, our recommendation is to use Multinomial Naïve Bayes without ensemble techniques when determining validity of reviews, since it is less computationally costly, faster, and no significant difference in

classification performance was observed with the addition of ensemble techniques.

Unfortunately, labeled data that represents a real world spam review environment is very limited. Thus, future work should consider creation and testing of new data sets, as well as testing our experimental results on new data sets to see if they generalize.

VI. Acknowledgement

The authors gratefully acknowledge partial support by the National Science Foundation, under grant number CNS-1427536. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. L. Berenson, M. Goldstein, and D. Levine, *Intermediate Statistical Methods and Applications: A Computer Package Approach 2nd Edition*. Prentice Hall, 1983.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Jan 2001.

[3] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, Sept 1996. [Online]. Available: http://dx.doi.org/10.1007/BF00058655

[4] K. Buhrmester and Goslining, "Amazon's mechanical turk a new source of inexpensive, yet high-quality data?" *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, Jan 2011. [Online]. Available: http://pps.sagepub.com/content/6/1/3

[5] M. Crawford, T. M. Khoshgoftaar, and J. D. Prusa, "Reducing feature set explosion to facilitate real-world review spam detection," in *Proceedings of the 29th International FLAIRS conference*, May 2016, pp. 304–309.

[6] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *Journal Of Big Data*, vol. 2, no. 1, pp. 1–24, Dec 2015. [Online]. Available: http://link.springer.com/article/10.1186/s40537-015-0029-9

[7] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple classifier systems*. Springer, 2000, pp. 1–15.

[8] D. J. Dittman, T. M. Khoshgoftaar, and A. Napolitano, "Selecting the appropriate ensemble learning approach for balanced bioinformatics data." in *FLAIRS Conference*, 2015, pp. 329–334.

[9] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the 13th International Conference on Machine Learning*, Jan 1996, pp. 148–156.

[10] I. C. Government of Canada. (July 2014) Don't buy into fake online endorsements —not all reviews are from legitimate consumers. [Online]. Available: http://www.competitionbureau.gc.ca/eic/site/cb-bc.nsf/eng/03782.html

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: An update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[12] C.-W. Hsu, C.-C. Chang, and C.-J. Lin., "A practical guide to support vector classification," pp. 1–16, 2003.

[13] D. W. H. Jr and S. Lemeshow, *Applied Logistic Regression*. John Wiley & Sons, 2004.

[14] J. Li, O. Myle, C. Cardie, and E. Hovy, "Towards a general rule for identifying deceptive opinion spam," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, June 2014, pp. 1556–1576. [Online]. Available: http://anthology.aclweb.org/P/P14/P14-1147.pdf

[15] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, Jul 1998.

[16] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?" in *Seventh International AAAI Conference on Weblogs and Social Media*, June 2013. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6006

[17] M. Ott, Y. Choi, C. Cardie, and J. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, June 2011, pp. 309–319.

[18] J. D. Prusa, T. M. Khoshgoftaar, and D. J. Dittman, "Using ensemble learners to improve classifier performance on tweet sentiment data," in *2015 IEEE International Conference on Information Reuse and Integration (IRI)*, 2015, pp. 252–257.

[19] J. Prusa, T. M. Khoshgoftaar, and A. Napolitano, "Utilizing ensemble, data sampling and feature selection techniques for improving classification performance on tweet sentiment data," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2015, pp. 535–542.

[20] J. D. Prusa, T. M. Khoshgoftaar, and A. Napolitano, "Using feature selection in combination with ensemble learning techniques to improve tweet sentiment classification performance," in *Proceedings of the 27th International Conference on Tools with Artificial Intelligence*, Nov 2015, pp. 186–193.

[21] R. J. Quinlan, *C4.5: Programs for Machine Learning*. Elsevier, June, 2014.

[22] S. Shojaee, M. Murad, N. Sharef, and S. Nadali, "Detecting deceptive reviews using lexical and syntactic features," in *2013 13th International Conference on Intelligent Systems Design and Applications (ISDA)*, Dec 2013.

[23] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.