Comparison of Stability for Different Families of Filter-Based and Wrapper-Based Feature Selection

Randall Wald, Taghi Khoshgoftaar, Amri Napolitano
Florida Atlantic University
Email: {rwald1, khoshgof}@fau.edu, amrifau@gmail.com

***Abstract*—Due to the prevalence of high dimensionality (having a large number of independent attributes), feature selection techniques (which reduce the feature subset to a more manageable size) have become quite popular. These reduced feature subsets can help improve the performance of classification models and can also inform researchers about which features are most relevant for the problem at hand. For this latter problem, it is often most important that the features chosen are consistent even in the face of changes (perturbations) to the dataset. While previous studies have considered the problem of finding so-called "stable" feature selection techniques, none has examined stability across all three major categories of feature selection technique: filter-based feature rankers (which use statistical measures to assign scores to each feature), filter-based subset evaluators (which also employ statistical approaches, but consider whole feature subsets at a time), and wrapper-based subset evaluation (which also considers whole subsets, but which builds classification models to evaluate these subsets). In the present study, we use two datasets from the domain of Twitter profile mining to compare the stability of five filter-based rankers, two filter-based subset evaluators, and five wrapper-based subset evaluators. We find that the rankers are most stable, followed by the filter-based subset evaluators, with the wrappers being the least stable. We also show that the relative performance among the techniques within each group is consistent across dataset and perturbation level. However, the relative stability of the two datasets does vary between the groups, showing that the effects are more complex than simply "one group is always more stable than another group."**

***Keywords*-Stability, filter-based feature selection, wrapper-based feature selection**

I. Introduction

Many datasets face the problem of high dimensionality: having a very large number of features, especially in comparison with the total number of instances. These excess features can severely impact data mining in a number of ways: they can make computations much slower, can directly reduce the performance of classification models built on high-dimensional datasets, and can make it difficult for researchers and practitioners to understand the models and the role played by the features. To resolve these concerns, a collection of techniques known as feature selection have been developed to remove features which are irrelevant (not correlated with the class of interest) and redundant (containing information already found in other features).

Feature selection techniques fall into three major categories: filter-based feature ranking techniques, which use statistical measures to assign a score to each feature and then rank the resulting features based on these scores; filter-based subset evaluation, which uses a search technique to explore different possible feature subsets and then applies statistical techniques to each to find their merit; and wrapper-based subset evaluation, which also considers whole subsets but evaluates their merit by building a classification model using just those features and considering the performance of said model. These three categories of techniques have a range of benefits and disadvantages: feature ranking is much quicker than the other techniques (since features need only be evaluated once, rather than considering a large collection of feature subsets), but is unable to detect redundant features, and thus often requires larger feature subset sizes. Conversely, the subset evaluation techniques both are significantly more computationally expensive, but can give smaller and more useful feature sets. Between these two, filter-based subset evaluation is usually quicker (as the statistical tests are less computationally-intensive than building a full classification model), but only wrapper-based techniques can find those features which will actually give the best performance for the classification model. As each group of techniques has its own advantages and disadvantages, each has its own place in feature selection.

While one major goal of feature selection is improving the performance of classification models, frequently these models are not the main end product of the data mining process: instead, the set of important features is what matters most. Since the features themselves are the product, it is required that the chosen features not change due to slight changes or perturbations in the dataset. So-called "stable" approaches are able to produce such feature lists, while "unstable" approaches will give different feature lists when the dataset is changed.

While the problem of stability has been considered for each group of feature selection techniques separately, few works have examined the stability of all three major categories using one dataset and one stability metric. That is the contribution of this paper: to find how these three groupings relate in terms of stability, and to discover what patterns arise from their stability results. To this end, we employed

978-0-7695-5144-9/13 $31.00 © 2013 IEEE
DOI 10.1109/ICMLA.2013.162

two datasets from the domain of mining Twitter profiles to discover susceptibility to social bots, along with two levels of dataset perturbation (e.g., variation applied to the dataset to generate the randomness necessary for evaluating stability). We compared five filter-based feature ranking techniques, two filter-based subset selection techniques, and five learners used within wrapper-based subset evaluation, with all stability values evaluated using the Average Tanimoto Index (ATI) as suggested by Kalousis et al. [10].

Our results show that filter-based feature ranking is much more stable than the other two groups: no feature ranker has a stability below 0.25 ATI for either dataset or for the two choices of perturbation level (67% and 80%), while only one other result (using the Consistency filter-based subset evaluator with one of the datasets at 80% perturbation) exceeds this value. Furthermore, we find a consistent ordering of techniques within each grouping, although some slight variations exist. We also find it notable that the relative order of the two datasets (in terms of stability) varies depending both on the perturbation level examined and the group of feature selection techniques used — suggesting that properties of the dataset have a subtle but real effect on the stability of different techniques, and that not all techniques respond to these properties in the same way. Overall, we find that only the filter-based feature ranking family provides reasonably stable feature subsets, but that within all three groupings, a clear order (in terms of stability) is present.

The remainder of this paper is organized as follows: Section II presents related work on the three categories of feature selection and on our application domain. Section III discusses the methods used in this work, including the feature selection and stability measurement approaches. In Section IV, we illustrate our case study to show where our data comes from. Section V contains our results. Finally, in Section VI we conclude and discuss ideas for future work.

II. Related Work

Feature selection is an extremely important problem across a wide range of application domains: any time a dataset has too many features, or practitioners would like to know which features are most important, feature selection can provide a reduced list with just those that matter most. A broad survey of this field is presented by Guyon and Elisseeff [5], who divide feature selection techniques into two broad categories: filters and wrappers. Filters are defined as any technique which uses statistical methods to find the best features from the dataset; these can score each feature individually and then rank features based on their scores, or calculate a metric from a whole feature subset to describe the goodness of that subset. Wrappers, on the other hand, incorporate learners to determine which feature subsets are actually best for building models. Embedded techniques are noted as a special case of wrapper selection, where instead of using the learner to select features and then using it again to build a model, the feature selection is embedded directly in the model-building process.

Feature selection has been heavily studied for many years [13], but although feature ranking techniques have been widely examined, the increased complexity of filter-based subset evaluation and wrapper-based subset selection techniques has resulted in these being less well-studied [12]. Even fewer works directly compare three different categories of feature selection (filter-based ranking, filter-based subset evaluation, and wrapper-based subset selection), and those that do are relatively unsophisticated: for example, Molina et al. [16] use rankers which later research has shown to be ineffective [2], and the chosen subset search techniques are also fairly simple. In contrast, Gheyas and Smith [4] propose a hybrid technique incorporating both filter and wrapper-based ideas, and compare this with other filter-wrapper hybrids, but do not consider either filters or wrappers alone. No previous works consider the previous decade of advancement in the area of feature selection to determine how feature ranking, filter-based subset evaluation, and wrapper-based subset selection compare with one another when using the best-known examples of each family.

While feature subset selection has received relatively little attention, evaluating the stability of these techniques has received even less focus. He and Yu [8] reviewed causes of instability and stability metrics, including metrics which may be applied towards feature subset selection. Somol and Novovičová [18] also evaluated stability metrics, using both simulated and real data to observe how different metrics can give different results as well as how three wrapper-based techniques (using a Bayesian classifier, 3-Nearest Neighbor, and Support Vector Machines) compare to one another. Yu et al. [28] proposed both a new filter-based feature subset evaluation technique and a new stability metric for evaluating such techniques, both based on the idea of feature groups (selecting a collection of feature subsets, where the features within each subset are expected to be redundant with one another, rather than simply selecting individual features). Lustgarten et al. [14] propose a new stability metric for feature subset evaluation (based on Kuncheva's consistency index), and compare this metric with the Jaccard index using three wrapper-based subset selection techniques (Logistic Regression, Naïve Bayes, and SVM). Dunne et al. [3] consider wrappers using a 3-nearest neighbor learner and three choices of search technique, evaluating stability by resampling the original dataset and finding the Hamming distance between the various feature subset masks. Haury et al. [7] evaluate a number of different feature selection techniques (primarily rankers, but including one wrapper-based subset evaluation technique using least squares regression) and consider stability in terms of how many features are in common between two subsets generated from independent subsamples of the original data. Overall, no work has considered both filter-based subset evaluation and wrapper-based subset selection at the same time, or has examined a wide range of both learners and performance metrics within the context of wrapper-based feature selection.

The large number of users on Twitter makes it a major target for agencies seeking to create buzz for a product or service. In particular, automatically-generated advertising messages have become a significant problem for Twitter users [23]. This has led to research focused on detecting spam on Twitter, using techniques including traditional classifiers [15], considering the network relationships between the sender and receiver [19], and evaluation of the URLs being promoted by the spammers [22].

In 2011, the Web Ecology Project began their Socialbot Challenge [9] to promote the study of Twitter social bots. Three teams created bot clusters in order to elicit real users to follow and reply to their bots, accruing points based on how many users interacted with the "lead" bot on each team. Over the course of the two-week challenge, the top team was able to accumulate approximately 8 followers per day and 14 replies per day, with the latter metric far outweighing the other two competitors. An unaffiliated set of researchers, Wagner et al. [25], realized that the data from this challenge could help understand users who choose to follow bots. Three categories of features were extracted from each user to predict their susceptibility to bots: 70 linguistic features (which used the Linguistics Inquiry and Word Count (LIWC) [21] package to extract word-use dimensions from users' tweets), nine network-based features using three different forms of graph generation (a follower-based directed graph, an undirected retweet-based graph, and a raw interaction graph), and 13 behavioral features based on the scope and range of tweet contents. Using this dataset and six classification learners, Wagner et al. were able to achieve an overall accuracy of 0.71, and a closer analysis of the features shed light onto the psychological traits leading to bot susceptibility.

III. Methods

In this work, three forms of feature selection are considered: filter-based feature ranking, filter-based feature subset evaluation, and wrapper-based subset selection. In addition, feature selection stability is evaluated using the perturbation framework and with the ATI metric. These methods are discussed in detail below.

A. Feature Selection

Three different approaches to feature selection are considered in this work: filter-based feature ranking, filter-based subset evaluation, and wrapper-based subset selection. All three share the general pattern of "evaluate elements of the feature space and find the best solutions," but the exact mechanism of this differs between the feature rankers and the subset evaluators. With feature ranking, the process is simple: each feature is scored individually, and the top N features are used. More information on this is found in Section III-A1.

With the two subset evaluation-based groups, though, a search technique must be used to explore the space of all possible feature subsets, to reduce the problem from being $O(2^n)$. In all subset evaluation experiments in this paper, we used greedy forward selection. In this process, we start with the empty set as our "candidate" set and consider all sets which contain the candidate set and exactly one feature not currently in the candidate set. These are evaluated (either with a statistical measure in filter-based subset evaluation or through building a classifier in wrapper-based subset evaluation), and the best feature is added to the candidate set. The process is repeated until none of the potential new candidate sets show improved performance over the previous candidate set. The specific details regarding the filter and wrapper-based versions of subset evaluation are found in Sections III-A2 and III-A3, respectively.

1) Filter-Based Feature Ranking: For feature ranking, we choose five representative techniques: Deviance (Dev), Mutual Information (MI), Area Under the Precision-Recall Curve (PRC), Signal-To-Noise (S2N), and Significance Analysis of Microarrays (SAM). These were chosen for two reasons. First of all, they represent two major groupings of feature ranking technique: Dev, MI, and PRC are examples of threshold-based feature selection (TBFS), while S2N and SAM are examples of first-order statistics-based feature selection (FOS). In addition, these five techniques can be divided into high-stability (Dev, SAM, PRC) and low-stability (S2N, MI) groups based on preliminary investigation. We wanted to include a diverse range of feature selection approaches (both in terms of technique and stability performance) to give a broader view on this form of feature selection and to give a more full comparison with other forms of feature selection. Due to space limitations, only a brief description of each technique is presented below; further information on TBFS techniques may be found in [24], while further information on FOS-based methods may be found in [11]. All techniques were implemented within the WEKA machine learning framework [27] by our research team.

Threshold-based feature selection consists of considering each feature in isolation with the class variable: effectively, a new, two-feature dataset is built pairing each feature with the class value. Then, the value of the given feature is normalized to lie between 0 and 1, and this normalized value is considered as a posterior probability, as though it were the output of a classification model. However, since no actual model is built, this is still a filter-based method. Considering these values as posterior probabilities, a threshold is chosen, and values above this threshold are placed into one class (with those below it placed into the other class). This threshold is varied from 0 to 1, and different classification performance metrics (in the present work, the Dev, MI, and

PRC metrics) are used to judge how well the predicted class values match with the actual class values. The optimal choice over all threshold levels (and over the choice of making high or low feature values correspond to positive-class instances) is used to represent the performance of that feature.

First-order statistics based feature selection is a family of related techniques which all center around the use of first-order statistics such as mean and standard deviation. S2N is a technique in this family which considers the ratio of the "signal" from the feature, here defined as the difference in the feature's mean values for the two classes, divided by the "noise" of the feature, which is the sum of the feature's standard deviation for the two classes. SAM is a technique originally developed for bioinformatics work which also has the difference of means as the numerator of the expression, but which uses the square root of the normalized sum of the variances from the two classes (added to a small regularization constant) as its denominator.

For all feature ranking-based approaches, we chose the top 40 ranked features for evaluating stability. This number was based on preliminary research suggesting it was appropriate for this experiment. Note that this does imply using more features for the rankers than were used in subset evaluation (because subset evaluation techniques choose their own stopping point, usually with fewer than 25 features), but it is difficult to compare directly as the number of features for subset evaluation will vary in the course of our experiments (for example, with different datasets and iterations within the cross-validation process). In addition, it would be unfair to intentionally misuse feature ranking by reducing the number of features too much in an attempt to match the feature sets chosen by subset evaluation-based methods.

2) Filter-Based Subset Evaluation: Two forms of filter-based subset evaluation are used to find the quality of the various feature subsets. These were chosen due to being the most widely-used in the literature, as well as being the only two with implementations in WEKA. The first of these is Correlation-Based Feature Selection (CFS) [6]. This employs the Pearson correlation coefficient, a correlation metric designed to balance the need to have the features correlate with the class and the need to have the features not correlate with one another. The Pearson correlation coefficient is found with the following formula:

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

In this formula, M_S is the merit of the current set of features, k is the number of features, $\overline{r_{cf}}$ is the mean of the correlations between each feature and the class, and $\overline{r_{ff}}$ is the mean of the pairwise correlations between every two features. In both cases, correlations are calculated using symmetric uncertainty, an information-theoretic measure of how changes in one feature affect the uncertainty of the other, and which compensates for inherent entropy in either feature. As desired, the numerator increases when the set of features is particularly good at classifying the data, while the denominator increases when the set has a great deal of self-correlation, which implies redundancy.

The second filter-based technique used is Consistency [1], which seeks to find the largest possible feature subset which is "consistent." This is most easily understood in terms of its converse, inconsistency: a feature subset is considered "inconsistent" if two instances share all the same values for the given features but differ in their class variables. Mathematically, the inconsistency of a feature subset is found by first going through the dataset and finding all unique "patterns" produced by the current feature mask. A pattern is a tuple of feature values, considering only the features in the current feature set and not including the class attribute. The total number of instances which match a given pattern is represented by n, while the number of instances in classes C_1, C_2, C_3, etc. are represented by c_1, c_2, c_3, and so on. Without loss of generality, assume that class C_3 has the most instances for the chosen pattern, and thus c_3 is the largest of the c's; in this case, the inconsistency count of the pattern is found by $n - c_3$. To find the inconsistency count of the whole dataset, the individual counts from each pattern are added together, and the final result is divided by the total number of instances in the original dataset. Lower values of inconsistency count are preferred, because these have greater consistency.

3) Wrapper-Based Feature Selection: The basic premise of wrapper feature selection is building a model using a potential feature subset and using the performance of this model as a score for the merit of that subset [12]. As with any model-building process, a number of choices must be made in how to build and evaluate the model. First of all, while this model could be built using the full training set and then evaluating its performance against that same training set, this would potentially lead to overfitting: building models which memorize the data rather than learning general properties of the data. Thus, for our experiments the wrapper process uses cross-validation: the training set is divided into five parts, and models (using the potential feature subsets) are built on only four parts, and evaluated on the fifth. This process is repeated (by changing which folds are used for building the model) until all folds have been the evaluation fold exactly once, and the results are averaged to give the merit of the potential feature subset.

In this paper, five diverse learners are used both inside the wrapper and for final classification: 5-Nearest Neighbor (5-NN), Logistic Regression (LR), Multi-Layer Perceptron (MLP), Naïve Bayes (NB), and Support Vector Machines (SVM). Because these are each well-understood techniques, we provide only a brief discussion of how they predict class labels; an interested reader may consult Witten and Frank [27] for further information. All models were built using the WEKA machine learning toolkit [27], using the default parameter values unless otherwise noted.

5-Nearest Neighbor classifies instances by finding the five closest instances to the test instance and comparing the total weight of the instances from each class (using 1/Distance as the weighting factor). Logistic Regression builds a simple logistic model using all of the features in order to predict the class variable. Multilayer Perceptron builds an artificial neural network with three nodes in its single hidden layer, and 10% of the data held aside in order to validate when to stop the backpropagation procedure. Naïve Bayes uses Bayes' Theorem to determine the posterior probability of membership in a given class based on the values of the various features, assuming that all of the features are independent of one another. Finally, Support Vector Machines finds a maximal-margin hyperplane which cuts through the space of instances (such that instances on one side are in one class and those on the other side are in the other class), choosing the plane which preserves the greatest distance between each of the classes. In this paper, for SVM the complexity constant "c" was set to 5.0 and the "buildLogisticModels" parameter was set to "true."

While the choice of performance metric is important in any data mining study, this is especially important for considering wrapper-based feature selection, as the wrapper itself will use this performance metric to grade the subsets. The presence of imbalanced data also highlights the importance of this choice, in order to ensure that minority-class instances (also known as positive instances) do not all end up misclassified [17]. For this reason, we use Area Under the Receiver Operating Characteristic Curve (AUC) as our metric inside the wrapper. AUC builds a graph of the True Positive Rate vs. True Negative Rate as the classifier decision threshold is varied, and then uses the area under this graph as the performance across all decision thresholds.

B. Perturbation and Stability Measurement

In order to measure stability, we need both a system for producing the diversity which will create the various feature subsets for comparison, as well as a measure to actually compare these subsets. For this experiment, the former is accomplished through perturbation. A fraction of the instances (either 80% or 67%) from the original dataset are randomly chosen (without replacement), and this procedure is repeated 30 times to create 30 subsamples of the original data per perturbation level. These will all contain a slightly different view on the original data, to help determine how stable the techniques are in the face of such changes to the data; larger degrees of perturbation (where a smaller fraction of instances are chosen from the original data) will lead to greater diversity. Following the subsampling process, feature selection is then applied independently on each of these 30 subsamples, creating 30 feature subsets. These 30 feature subsets are then compared pairwise, resulting in $(30 \cdot 29)/2 = 435$ pairings, and the stability measure is used to find a score for each pairing; the average value is taken as the stability measure for the full collection.

The stability measure we use in this paper is the Average Tanimoto Index, derived from work originating in Kalousis et al. [10]. This metric was chosen due to its ability to measure the stability of a feature selection technique even if this technique does not consistently give the same feature subset size on different permuted datasets (as is the case with our subset evaluation techniques). The original Tanimoto Index defines the similarity between two feature subsets as follows:

$$\begin{aligned} S_K(S_i, S_j) &= \frac{|S_i \cap S_j|}{|S_i \cup S_j|} = 1 - TD(S_i, S_j) \\ &= 1 - \frac{|S_i| + |S_j| - 2|S_i \cap S_j|}{|S_i| + |S_j| - |S_i \cap S_j|} \end{aligned} \tag{1}$$

where $TD(S_i, S_j)$, which is clarified on the second line, is the Tanimoto distance between the two feature subsets.

Because the Tanimoto Index does not imply or require that subsets S_i and S_j have the same size, extending it to apply to arbitrary pairs of subsets is trivial, and thus the Average Tanimoto Index may be used to determine the stability of a single collection of feature subsets:

$$ATI(\mathcal{S}) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} S_K(S_i, S_j) \tag{2}$$

In this equation, S_i will be any subset in the collection other than the final subset, and S_j will be one of the subsets which follows S_i (including possibly the final subset); thus, all pairs of subsets within $\mathcal{S}$ will be considered once.

For the present work, we report the ATI value from comparing all 30 feature subsets generated by applying perturbation 30 times and building one feature subset from each perturbed version of the original dataset.

IV. Case Study

The case study for this experiment uses data prepared by the Online Privacy Foundation[1], an organization dedicated to understanding how users interact with social networks and the privacy implications of their actions. In a previous experiment, called the Twitter Big Five Experiment [20], [26] (relevant because it uses almost the same independent features as the present work, although the class value differs), a number of users were solicited to take an online personality survey. In addition, three classes of features were extracted from each individual's profile: demographic features, linguistic features, and count-based features.

Demographic features consisted of numeric information which could be directly extracted from the profile, or generated automatically using this numeric content: this includes facts like the number of friends and followers,

[1] https://www.onlineprivacyfoundation.org/

the number of statuses posted by the user, the length of their self-description, and so on. One demographic feature of special note is an individual's Klout score. This is a measure created by an independent company, Klout.com, to evaluate a person's overall reach in terms of social network connections. In general, a user's Klout score will depend on their accounts on many sites, including Twitter, Facebook, Google+, LinkedIn, and others. In the original Twitter Big Five Experiment, 20 demographic features were used, but for the present social bot experiment, there were 22.

The next category, linguistic features, was extracted using the Linguistic Inquiry and Word Count package [21], which divides words into 68 topical collections to represent different types of language and word use, and then counts how often a given individual uses words in each collection. In addition, 12 forms of punctuation are counted to evaluate how these reflect on a user's writing style. These counts were collected across all of a user's tweets, to generate an overall picture of their personal writing style and mental state. The tweets themselves were also divided into six separate groups: all tweets, tweets originating from the user, retweets (copies from other users), replies to other users, and tweets from days when the user tweeted more than either 10 or 40 times. A total of 80 linguistic features (68 topical + 12 punctuation) were extracted for each user per group of tweets, resulting in 480 features overall.

The third category, count-based features, consider the number of days on which a given user posts a certain number of tweets from a certain subcategory. For example, one such feature might be the number of days when a user posted more than 40 original tweets, or more than 10 retweets. In addition, the maximum and average number per day within the given subcategory are considered, along with the total number of days with one or more relevant tweet and the total number of such tweets. Overall, six subcategories of tweets (all tweets, original tweets, retweets, replies, tweets mentioning "Follow Friday" or the hashtag "#FF", and tweets which include the derogatory term "cunt") are considered. Because there are seven thresholds used for the first four subcategories (and eleven for the last two subcategories), plus the four maximum/average/total days/total tweets features per subcategory, there are a total of 74 count-based features.

Following the Twitter Big Five Experiment, a second experiment was performed on the same users, to study their response to social bots. The original pool of subjects was reduced to 610 users, who were then divided into two groups to be studied separately. However, the procedure for both was identical (the only difference being the name of the bot account used, and one group being tested approximately one month prior to the other, to avoid the problem of many users discovering the bot and realizing its nature), and as such these groups are pooled for the present work. For both groups, a Twitter bot was created which performed two tasks: it would post tweets meant to be representative of what normal Twitter users would post, and it would ask specific questions of the users in the group being tested, using Twitter's @ syntax to send these messages at the target users. A user was considered to have interacted with the bot if they replied to this message or if they subsequently chose to follow the bot. Only users who were part of the original Twitter Big Five Experiment were considered, even if other users followed (or sent messages to) the bot of their own accord. Thus, for each user, the 576 features were paired with one of two binary class variables: whether or not that user interacted with the bot, or whether they replied to the bot (disregarding whether or not they followed it). Thus, two datasets were created: the Interacted dataset (using the "interacted or not" class variable) and the Replied dataset (using the "replied or not" class variable).

Feature Selection		Dataset			
		Interacted		Replied	
Category	Type	Perturbation Level			
		67%	80%	67%	80%
Filter-Based Feature Ranking	Dev	0.45996	**0.53052**	**0.50584**	**0.58756**
	MI	0.32050	0.39593	0.31656	0.42225
	PRC	0.41964	**0.51338**	0.38640	**0.52797**
	S2N	0.29381	0.40912	0.29125	0.38980
	SAM	0.38388	**0.52138**	0.39740	**0.55890**
Filter-Based Subset Eval	CFS	*0.13827*	0.24036	*0.12930*	0.23825
	Consistency	*0.09054*	0.34150	*0.14877*	0.20489
Wrapper-Based	5-NN	*0.04508*	*0.04384*	*0.05672*	*0.05664*
	LR	*0.06616*	*0.10805*	*0.07814*	*0.08841*
	MLP	*0.03126*	*0.05176*	*0.03158*	*0.04417*
	NB	*0.11061*	*0.16213*	*0.11109*	*0.14170*
	SVM	*0.02428*	*0.02809*	*0.01991*	*0.02210*

Table I: Stability results for all categories of feature selection, on two datasets and two perturbation levels

V. Results

Table I presents the stability results for all feature selection techniques in this paper, across both datasets and both dataset perturbation levels. The first two columns identify the specific technique being demonstrated: the first of these shows which of the three categories (filter-based feature ranking, filter-based subset evaluation, or wrapper-based feature selection) was used, while the second shows the specific type within the given category. The remaining columns show the stability (measured using the ATI metric) when using the two different datasets or two different perturbation levels. Values less than 0.2 have been printed in *italics*, while those greater than 0.5 have been printed in **bold**.

The most evident result we find is that feature ranking is significantly more stable than subset evaluation (either filter- or wrapper-based). Even at their worst, no feature ranker gives a stability value of less than 0.25, while only in one instance (using the Consistency filter-based subset evaluator with 80% perturbation on the Interacted dataset) does a subset evaluation technique exceed this value. In fact, all results using wrapper-based subset evaluation give stability values below 0.2, while all results which use 67% perturbation and the filter-based subset evaluators likewise are below 0.2. This result is not entirely surprising, given that feature ranking allows for redundant features while subset evaluation seeks to eliminate these; this redundancy can directly improve stability, as can the practice of including more features when employing feature ranking (which we have also applied here, to give a realistic assessment of all techniques). Nonetheless, these results underscore how if stability is a desired goal, only feature ranking techniques are appropriate techniques.

Additional results may be found by considering the different techniques within each group. As we specifically chose three particularly stable feature rankers (Dev, SAM, and PRC) and two unstable feature rankers (S2N and MI), it is unsurprising to find that these groups remain distinct in terms of their stability performance across the datasets and perturbation levels. However, minor variations do exist within each group (for example, contrasting the results of the two perturbation levels within the Interacted dataset), which demonstrates that optimal and pessimal choices may vary depending on experimental conditions. In addition, the range between the most and least-stable rankers shows that the choice of ranker can influence stability.

Within the subset evaluation results, we also see a low degree of variation, although some does exist. For example, although CFS is typically more stable than Consistency, the reverse is true for perturbation level 67% on the Replied dataset. Likewise, among the wrapper-based techniques, the stability order (from most to least stable) is {NB, LR, 5-NN, MLP, SVM} for three of the four scenarios, but when using 80% perturbation on the Interacted dataset, the 5-NN and MLP-based techniques switch places. These suggest that although the actual stability of subset-based techniques is much lower than for ranking-based techniques, the results remain self-consistent. This is especially notable given that subset evaluators may vary their number of features chosen based on the stopping criterion making different choices on different data, and yet this does not appear to result in greater variability among the stability results of the different feature selection techniques.

Finally, comparing between the four scenarios created by combining two datasets and two perturbation levels, we find some unusual results. For the filter-based feature rankers, typically the greatest stability with 67% perturbation will be found with the Interacted dataset (3/5 times), while the greatest stability with 80% perturbation is found with the Replied dataset (4/5 times). The reverse is true with wrapper-based subset evaluation, however: 67% perturbation is most stable with the Replied dataset (4/5 times) and 80% perturbation is most stable with the Interacted dataset (also 4/5 times). As for the third category of feature selection, we observe that the filter-based subset evaluation group only has two members, and thus it is hard to draw meaningful conclusions from this. Nonetheless, it is important to note that the relative stability of the two datasets can vary depending on both the perturbation level and the family of techniques being evaluated.

VI. Conclusion

In this paper we sought to compare the stability of three different forms of feature selection: filter-based feature ranking, filter-based subset evaluation, and wrapper-based subset evaluation. To this end, we employed two datasets from Twitter profile mining, considered two different dataset perturbation levels, and used the ATI stability metric. In total, we compared five filter-based feature ranking techniques, two forms of filter-based subset evaluation, and five learners used within the wrapper-based subset evaluation framework. We found that feature ranking is far more stable than either subset evaluation approach, and that filter-based subset evaluation is more stable than wrapper-based subset evaluation. We also found distinct patterns of stability within each group: from most to least stable, the filter-based rankers were {Dev, SAM, PRC, MI, S2N}, the filter-based subset evaluators were {CFS, Consistency}, and the wrapper-based subset evaluators were {NB, LR, 5-NN, MLP, SVM}. However, it should be noted that there were some slight variations in these patterns for different choices of dataset and perturbation level. More notably, the relative stability of the two datasets varied both depending on the perturbation level and on the group of feature selection techniques: for filter-based feature rankers, one dataset is most stable at 67% perturbation and the other at 80% perturbation, while for the wrapper-based subset evaluators, the two datasets swap places. This shows that while the overall patterns of stability are consistent between and within the three types of feature selection, characteristics of the datasets can lead to more subtle effects.

Future work may consider a wider range of feature ranking techniques along with additional choices of performance metric within the wrapper. In addition, more datasets may be considered to ensure that these results generalize to additional application domains.

References

[1] M. Dash, H. Liu, and H. Motoda, "Consistency based feature selection," in *Knowledge Discovery and Data Mining. Current Issues and New Applications*, ser. Lecture Notes in Computer Science, T. Terano, H. Liu, and A. Chen, Eds. Springer Berlin Heidelberg, 2000, vol. 1805, pp. 98–109.

[2] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and J. Van Hulse, "Comparative analysis of DNA microarray data through the use of feature selection techniques," in *Ninth IEEE International Conference on Machine Learning and Applications (ICMLA)*, December 2010, pp. 147–152.

[3] K. Dunne, P. Cunningham, and F. Azuaje, "Solutions to instability problems with sequential wrapper-based approaches to feature selection," *Journal of Machine Learning Research*, 2002.

[4] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognition*, vol. 43, no. 1, pp. 5 – 13, 2010.

[5] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[6] M. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, Hamilton, New Zealand, April 1997.

[7] A.-C. Haury, P. Gestraud, and J.-P. Vert, "The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures," *PLoS ONE*, vol. 6, no. 12, p. e28210, 12 2011.

[8] Z. He and W. Yu, "Stable feature selection for biomarker discovery," *Computational Biology and Chemistry*, vol. 34, no. 4, pp. 215–225, 2010.

[9] T. Hwang. (2011) Help robots take over the internet: The socialbots 2011 competition: Web ecology project.

[10] A. Kalousis, J. Prados, and M. Hilario, "Stability of feature selection algorithms: a study on high-dimensional spaces," *Knowledge and Information Systems*, vol. 12, no. 1, pp. 95–116, May 2007.

[11] T. M. Khoshgoftaar, D. Dittman, R. Wald, and A. Fazelpour, "First order statistics based feature selection: A diverse and powerful family of feature seleciton techniques," in *11th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, Dec. 2012, pp. 151–157.

[12] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, Dec. 1997.

[13] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*, ser. The Springer International Series in Engineering and Computer Science Series. Kluwer Academic Print on Demand, 1998.

[14] J. L. Lustgarten, V. Gopalakrishnan, and S. Visweswaran, "Measuring stability of feature selection in biomedical datasets," in *AMIA 2009 Annual Symposium Proceedings*, 2009, pp. 406–410.

[15] M. McCord and M. Chuah, "Spam detection on Twitter using traditional classifiers," in *Autonomic and Trusted Computing*, ser. Lecture Notes in Computer Science, J. M. A. Calero, L. T. Yang, F. G. Mármol, L. J. García Villalba, A. X. Li, and Y. Wang, Eds. Springer Berlin Heidelberg, 2011, vol. 6906, pp. 175–186.

[16] L. Molina, L. Belanche, and A. Nebot, "Feature selection algorithms: a survey and experimental evaluation," in *Proceedings of the 2002 IEEE International Conference on Data Mining*, 2002, pp. 306–313.

[17] N. Seliya, T. M. Khoshgoftaar, and J. Van Hulse, "A study on the relationships of classifier performance metrics," in *21st International Conference on Tools with Artificial Intelligence*, November 2009, pp. 59–66.

[18] P. Somol and J. Novovičová, "Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 1921–1939, 2010.

[19] J. Song, S. Lee, and J. Kim, "Spam filtering in Twitter using sender-receiver relationship," in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science, R. Sommer, D. Balzarotti, and G. Maier, Eds. Springer Berlin Heidelberg, 2011, vol. 6961, pp. 301–317.

[20] C. Sumner, A. Byers, R. Boochever, and G. Park, "Predicting dark triad personality traits from Twitter usage and a linguistic analysis of tweets," in *11th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, 2012, pp. 386–393.

[21] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: Liwc and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24–54, 2010.

[22] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2011, pp. 447–462.

[23] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended accounts in retrospect: an analysis of twitter spam," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 243–258.

[24] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "A comparative evaluation of feature ranking methods for high dimensional bioinformatics data," in *2011 IEEE International Conference on Information Reuse and Integration (IRI)*, August 2011, pp. 315–320.

[25] C. Wagner, S. Mitter, C. Körner, and M. Strohmaier, "When social bots attack: Modeling susceptibility of users in online social networks," *Making Sense of Microposts (# MSM2012)*, p. 2, 2012.

[26] R. Wald, T. M. Khoshgoftaar, A. Napolitano, and C. Sumner, "Using Twitter content to predict psychopathy," in *11th International Conference on Machine Learning and Applications (ICMLA)*, vol. 2, 2012, pp. 394–401.

[27] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical machine learning tools and techniques*, 3rd ed. Burlington, MA: Morgan Kaufmann, January 2011.

[28] L. Yu, C. Ding, and S. Loscalzo, "Stable feature selection via dense feature groups," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '08. New York, NY, USA: ACM, 2008, pp. 803–811.