Recursive Feature Elimination by Sensitivity Testing

Anonymous Authors[1]

Abstract

There is great interest in methods to improve human insight into trained non-linear models, such as support vector machines (SVMs), deep neural networks, and large random forests; leading approaches include producing a ranking of the most relevant features, a non-trivial task for non-linear models. We show theoretically and empirically the benefit of a novel version of recursive feature elimination (RFE) as often used with SVMs; the key idea is a simple twist on the kinds of sensitivity testing employed in computational learning theory with membership queries (e.g., Bshouty et al. 1992). With membership queries, one can check whether changing the value of a feature in an example changes the label. In the real-world, we usually cannot get answers to such queries, so our approach instead makes these queries to a trained (imperfect) non-linear model. Because SVMs are widely used in bioinformatics, our empirical results use a real-world cancer genomics problem; because ground truth is not known for this task, we discuss the *potential* insights provided. We also evaluate on synthetic data where ground truth is known.

1. Introduction

There is great interest in methods to improve human insight into trained non-linear models such as support vector machines (SVMs), deep neural networks, and large random forests; one existing approach is to produce a ranking of the most relevant features, a non-trivial task for non-linear models. Famous examples of this approach include Breiman's method for ranking features in a random forest by lost area under the Receiver Operating Characteristic curve (AUC) when a feature is deleted (Breiman, 2001) and Guyon's modified version of her recursive feature elimination (RFE) approach tailored to non-linear models (Guyon et al., 2002), which both ranks and performs feature selection. Breiman's and Guyon's approaches test how much the deletion of a feature costs in lost accuracy or AUC. In contrast, computational learning theory has a long history of *sensitivity* testing by "flipping" or changing the value of a feature, rather than deleting it, and posing a membership query to find the effect on the label of the example (e.g., Bshouty et al. 1992). In practice we do not have an oracle for such membership queries.

This paper presents an alternative algorithm, RFE by Sensitivity Testing (RFEST), that employs a trained non-linear model as an approximate oracle for such membership queries. Hence our algorithm asks how much accuracy or AUC is lost from a *trained* model when a variable is *flipped*, rather than how much is lost *compared to* an existing model when a variable is *deleted*. We first prove a probably approximately correct (PAC)-like result showing that under certain assumptions this algorithm provides an accurate ranking; this result does not rely on any particular type of non-linear model or learning algorithm, but only on the condition that the algorithm achieves some minimum gain in accuracy over random guessing, as in weak learning. Second, we show empirically that when used with SVMs the new algorithm, RFEST, outperforms RFE in ranking (as a surrogate for insight) the genetic features associated with breast cancer in a genome-wide association study (GWAS) data set and on multiple synthetic data sets labeled by known ground truth, a family of arguably the most challenging non-linear target functions. Our study is limited to binary data but the algorithm potentially can be extended to other types of data by permuting feature values.

As a motivating example, genome-disease association studies (genome-wide or limited) seek genetic features associated with disease, i.e., predictive of disease. In many cases it is believed that such features may interact with one another in highly nonlinear ways to influence disease; nevertheless, for practical reasons almost all association studies use linear models and hence can find only features that individually are correlated with disease (Zhang et al., 2010). Consequently, key genetic features may be missed entirely. Because linear and non-linear SVMs have been widely used in bioinformatics applications, we will use SVMs as our learner for these empirical studies. The theoretical results show the algorithm

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

can use any learner capable of building moderately accurate non-linear models.

2. Background

2.1. Feature Ranking and Feature Selection

While feature ranking and feature selection are different problems, they are closely related and each is sometimes accomplished by the other. Recursive feature selection can rank by maintaining the order in which features are removed; feature ranking, e.g. by information gain, is often followed by removal of lower ranked features. Many feature selection algorithms utilize linear modeling approaches such as lasso-penalized logistic regression, linear SVMs, naïve Bayes or other weighted-voting schemes among features. An alternative is to implement these same approaches after filtering features individually by information gain or by many single-variable logistic regression runs (Chandrashekar & Sahin, 2014). To account for interactions between features, the standard approach is to introduce interaction terms, but such terms typically are limited to pairs of features, and even then they greatly increase both run-time and risk of over-fitting.

Nonlinear SVMs have the potential to more effectively find complex interactions among features, but insight into the important interactions is hard to extract from the learned model. This paper addresses that shortcoming by presenting an alternative RFE algorithm and demonstrating that the algorithm makes it possible to identify the features that are relevant—that play a role in the learned nonlinear model, even if individually they are completely uncorrelated with the class—while removing those features that are irrelevant or redundant.

Because we do not assume we are in an active learning setting—we do not have access to an oracle for membership queries that can label feature vectors with the value of any feature altered— our key insight is to use the trained nonlinear SVM itself as such an oracle instead. While this trained model is not the target concept, we assume it is more accurate than random guessing and hence provides some information about feature relevance. In homage to earlier work on membership queries to test the sensitivity of target concepts to individual features, we call our RFE algorithm "RFE by Sensitivity Testing," or RFEST. The remainder of the paper presents the RFEST algorithm and empirical evaluations of it, including novel and promising insights that it provides into genetic susceptibility to breast cancer.

2.2. SVMs, Correlation Immunity, and RFE

SVMs are known for both their strong performance and flexibility based on the chosen kernel (Caruana & Niculescu-Mizil, 2006). The strength of SVMs comes from their ability to effectively learn nonlinear separators through use of the kernel trick, a mapping to a higher-dimensional feature space resulting in an ability to encode nonlinear separators in the original feature space (Maldonado & Weber, 2011).

Accordingly, it is expected that SVMs can efficiently learn correlation immune (CI) functions, which are notable non-linear Boolean functions. A function is CI if every single-feature marginal distribution is uninformative, i.e., no feature by itself is correlated with the function value, or class, even given the entire truth table or example space. We say a function f is *correlation immune of order c* (or c-correlation immune) if f is statistically independent from any subset of variables with a size of at most c. A function is correlation immune if and only if every variable has zero gain (with respect to any gain measure) when computed from the input data (cf. Roy, 2002).

Table 1. A truth table for Drosophila (fruitfly) survival based on gender and Sxl gene activity.

GENDER FEMALE	SXL ACTIVE	SURVIVAL
0	0	0
0	1	1
1	0	1
1	1	0

As a result, these functions include some of the most challenging target concepts for most classification algorithms, most noteworthy the parity functions. The most famous nonlinear separators in machine learning are exclusive-or (XOR) and exclusive-nor (XNOR), which are two-feature parity functions. These particular functions arise in practice, for example in biology (Table 1) (Cline, 1979). In Table 1, the interpretation of this output is that flies that will survive are either male with an active *Sxl* gene, or female with an inavtive *Sxl* gene. While a nonlinear SVM can learn this function easily given only the relevant variables (i.e. *Gender Female* and *Sxl Active*), the SVM's accuracy will degrade dramatically as irrelevant variables are added, unless the training set is quite large (see **Experimental Results**).

One would expect, for example, that SVMs, with a Gaussian kernel or polynomial kernel of degree at least two, would learn these functions with ease. Unfortunately, for the simplest case of XOR in the presence of even a modest number of irrelevant features, or variables, SVMs tend to have a difficult time learning and require a large sample size empirically. This problem is not specific to SVMs; it is known that no algorithm based on statistical queries can PAC learn parity functions of $log(n)$ variables (Blum et al., 1994). We seek a method of feature selection that can remove the irrelevant variables and restore classification performance.

A widely used approach to perform such a task is RFE, an embedded-based backward selection strategy (Stambaugh et al., 2013). RFE constructs an SVM, ranks the features ac-

cording to the constructed SVM, removes the lowest ranked feature or features (e.g., bottom ten percent), and repeats until a certain (user-specified) number of features remain. The RFE algorithm with a linear SVM simply ranks features with respect to their given coefficients (i.e. from the learned model); this approach assumes features have been normalized to have comparable ranges.

Unfortunately, for a nonlinear SVM, feature coefficients cannot be obtained; Guyon *et al.* (Guyon et al., 2002) presented a version of RFE for use with nonlinear SVMs. We propose an alternative RFE algorithm, RFEST, and compare these algorithms on synthetic data and a real-world cancer genomics problem.

2.3. Breast Cancer and Single-Nucleotide Polymorphisms (SNPs)

The development of breast cancer is influenced by many genetic and environmental factors. We study how feature selection performs on the variations at single base pairs of the human genome, which are known as single-nucleotide polymorphisms (SNPs). In cancer, both germline SNPs (the DNA sequence with which a person is born, and which is replicated in most of the cells in her body) and somatic mutations (variants that occur in select cells during replication and can lead to cancer) are important and are widely studied. To date, germline SNPs have received more attention as they can predict a person's future risk of breast cancer (Michailidou et al., 2017). Genome Wide Association Studies (GWAS) seek to find SNPs that are associated—correlated—with risk for developing disease.

Currently, GWAS consider SNPs independently and do not take into account possible interactions between SNPs. The rationale behind this is that it is infeasible, for example, to consider all pairs of the $n = 1\ million$ SNPs that are typically measured. The main purpose of a thorough investigation of SNPs is to gain a better understanding of how these genetic variants act as biological markers. Given a set of SNPs, if we can help identify a subset of important SNPs that correlate with a particular effect in patients, then we will be able to investigate their interactions. In turn, this will help our decision processes about numerous aspects of medical care such as the following: risk of developing a certain disease, effectiveness of various drugs, and adverse reactions to specific drugs.

3. Algorithms

3.1. RFE Algorithm

In our experiments, we compare our RFEST algorithm to the RFE method proposed by Guyon *et al.* (Guyon et al., 2002). Although variants of RFE have been proposed (Nguyen & de la Torre, 2010; Wang et al., 2010; Liu et al., 2011), the original method of Guyon *et al.* is still widely used in the bioinformatics community (Tao et al., 2015; Schwartz et al., 2015; Qureshi et al., 2016; Zarogianni et al., 2017; Kampe et al., 2017; Kong et al., 2017). Due to the nature of the data sets used in their paper, Guyon *et al.* utilized a linear SVM with RFE. However, they described how their method can be carried over to handle a nonlinear SVM implementation and this is the algorithm that we use as the baseline, which we describe next.

For SVMs, the cost function that is being minimized is the following:

$$J = \frac{1}{2}\boldsymbol{\alpha}^{\mathbf{T}}\boldsymbol{H}\boldsymbol{\alpha} - \boldsymbol{\alpha}^{\mathbf{T}}\mathbf{1} \tag{1}$$

with the following constraints:

$$0 \leq \alpha_k \leq C, \quad \sum_k \alpha_k y_k = 0$$

For training instances c and d, $\boldsymbol{\alpha}$ is a vector of weights on the training instances learned by the SVM algorithm, y_c and y_d represent the class values for the training instances, $\mathbf{x}_c$ and $\mathbf{x}_d$ are the feature vectors for the training instances, C is a regularization parameter, and $\boldsymbol{H} = y_c y_d K(\mathbf{x}_c, \mathbf{x}_d)$, for K a kernel function (Guyon et al., 2002).

To determine feature relevance, the change in cost function proposed by Guyon is the following ranking coefficient:

$$DJ(j) = \frac{1}{2}\boldsymbol{\alpha}^{\mathbf{T}}\boldsymbol{H}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^{\mathbf{T}}\boldsymbol{H}(-j)\boldsymbol{\alpha} \tag{2}$$

where $\boldsymbol{H}(-j)$ represents a modified version of $\boldsymbol{H}$ that recomputes the matrix without the j^{th} feature. In turn, the feature with the smallest value for $DJ(j)$ is removed.

Algorithm 1 RFE Algorithm

Input: data $d_{i,j}$, where $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$
repeat
  Train SVM, output $\boldsymbol{\alpha}$
  Implement $DJ(j)$ according to (2), $\forall$ features j
  Remove the feature(s) with the smallest $DJ(j)$
until k features remain ($k < n$)

Algorithm 1 describes the RFE algorithm for the nonlinear case in more detail (Lal et al., 2006). The benefit of using RFE over a vanilla approach (e.g. train a new SVM for each candidate feature on every iteration) allows for each iteration of the algorithm to train only one SVM model. In other words, we assume that the vector $\boldsymbol{\alpha}$ is fixed and consider the change in the kernel as a result of removing feature j. Note that for each iteration, $\boldsymbol{H}(-j)$ must be computed for each candidate feature j. The qualitative justification behind this cost function is that a feature's value to the learned model is measured by the change in the expected value of error when removing that candidate feature (Lal et al., 2006). RFE iterates until k features remain, however to have a fair

comparison to RFEST, the stopping criterion for RFE was set to be until the accuracy measurement AUC is less than the max AUC achieved thus far. We describe RFEST in the next section.

3.2. RFEST Algorithm

While our presentation of RFEST assumes that we are using binary features with a {-1,1}-encoding, it could be extended to handle continuous and/or categorical features as well. Standard RFE requires recomputing the $\boldsymbol{H}$ matrix (as described in the previous section) for each feature removed and can become computationally intractable with many features. In the case where there are thousands of features, Guyon *et al.* (Guyon et al., 2002) chose to remove half of the features at each iteration. Doing so allows for faster convergence to an idealized subset of features, but key information may be lost.

There are two main differences between RFE and RFEST. The first is that RFEST flips the binary features, rather than deleting them. Note that flipping a feature means that if its current value is -1, then it is changed to have the value 1, and vice versa.

The second difference is in the construction of the cost function. An SVM classifier can classify a dataset with the accuracy measurement AUC. In addition, for each feature j, we create a modified version of the training set by flipping feature j in each example, and then calculate the AUC of the same SVM classifier on this modified training set. We call the calculated value $AUC_{flipped}$. The ranking coefficient used by RFEST is the following:

$$R(j) = AUC - AUC_{flipped} \tag{3}$$

The interpretation of $R(j)$ is as follows. For each j, if $AUC_{flipped} < AUC$, then the j^{th} feature is relevant because the model classified the instances at a lower AUC with j flipped. In contrast, if $AUC_{flipped} \geq AUC$, then the j^{th} feature is irrelevant because the model classified the instances with the same or higher AUC with j flipped. Therefore, the feature corresponding to the smallest $R(j)$ will be removed. This process does not retrain a classifier for every candidate feature to be removed and we no longer compute $\boldsymbol{H}(-j)$.

For our experiments, a nonlinear SVM with an RBF kernel was used. The reason for doing so is because the RBF kernel implicitly computes interaction terms for all subsets of input features. In addition, it has been shown that searching in exponentially growing sequences for the hyper-parameters, namely cost C and gamma γ, is a good method for identifying their respective parameter values (Hsu et al., 2010). Therefore, the best configuration for C and γ was chosen using grid search.

To determine the final subset of features, RFEST stops when the AUC at any given iteration is less than $p\%$ of the max AUC achieved thus far. This parameter controls the trade-off between model interpretability and model efficacy. That is, a lower choice for p encourages a small number of coefficients, whereas a larger choice encourages a better performing model. For our experiments, we set $p = 95\%$ (see **Section 4**). Alternatively, other heuristics can be implemented with RFEST. One approach would be to stop when AUC decreases (i.e. a hill-climbing search). Another would be to apply a simulated annealing method, where if the AUC decreases, we continue searching with a small probability. As the search continues, the probability decreases. Search methods such as simulated annealing or the approach RFEST currently uses are aimed towards avoiding a local optimum. We use our current search method since it avoids having to set additional parameters.

RFEST may suffer if the SVM model we use is overfitted to a given instance, since the model might then be sensitive to the value of every feature. For this reason, we used a ten-fold cross validation and allocated the dataset into separate training, tuning, and testing sets to produce an unbiased estimate of the efficacy of our approach. The algorithm below summarizes RFEST.

Algorithm 2 RFEST Algorithm

Input: data $d_{i,j}$, where $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$
repeat
  Train SVM and output accuracy measurement AUC
  Implement $R(j)$ according to (3), $\forall$ features j
  Remove the feature(s) with the smallest $R(j)$
until AUC is less than $p\%$ of the max AUC achieved

We next demonstrate the theoretical efficacy of using the quantity $R(j)$ to rank features, by considering a classification problem on n binary features, where examples are labeled according to the parity of a subset of the features. We show if a nonlinear machine learning algorithm can learn a sufficiently accurate model M, then with high probability, using a polynomial-size sample to compute the $R(j)$ values with model M will result in those values being higher for relevant features than for irrelevant features. Thus if any irrelevant features are present, the feature with the lowest $R(j)$ value, removed by RFEST, will be irrelevant.

RFEST is a generic method. Because of the interpretation behind the ranking coefficient $R(j)$, this allows us to use any accuracy measurement. Therefore, for simplicity and clarity of our analysis, we prove our theorem for a related measure, $\tilde{R}(j)$, which is the same as $R(j)$ except that it is defined in terms of accuracy rather than AUC. Although we state the theorem here only for the parity function and uniformly distributed examples, we prove a more general theorem in the Supplementary Material. That theorem ap-

plies to a somewhat broader class of functions and product distributions.

Theorem 3.1. *Let f be a Boolean target concept, defined on n Boolean features, which labels examples according to the parity of a fixed subset of the n features. Suppose a machine learning algorithm is used to learn a classifier M for f. Suppose further that M has true error rate $\epsilon < 1/2$, with respect to the uniform distribution. Then there is a quantity t that is polynomial in $n, \ln \frac{1}{\delta}$, and $\frac{1}{(1/2)-\epsilon}$, with the following property: for all $0 < \delta < 1$, if the $\tilde{R}(j)$ values for all n features are computed using M and a new independent sample of size t, drawn from the uniform distribution, then with probability at least $1-\delta$, the computed $\tilde{R}(j)$ values for all the relevant features will be higher than the computed $\tilde{R}(j)$ values for the irrelevant features.*

While **Theorem 3.1** may appear to contradict the known result that parity functions are not PAC-learnable from statistical queries, it does not because it is preconditioned on having an SVM model with accuracy better than random guessing.

4. Experimental Results

4.1. Data

We implemented RFEST and Guyon's RFE algorithm tailored to a nonlinear SVM, and we evaluated it on two types of data with the programming language R (Karatzoglou et al., 2004; Kuhn et al., 2016; Wickham, 2011; Sing et al., 2005). The first consists of synthetic data that takes the form of a parity function on two variables, which is a CI function of order two. Correlation immune functions of order four, five, and six were also evaluated. There are many different CI functions, so for orders four, five, and six, ten functions for each order were randomly chosen. For functions of order c, the associated target concept was defined on n features. Of those n features, c were randomly chosen and corresponded to the c variables of the CI function, and the remaining features were irrelevant. Therefore, the task of both feature selection algorithms was to find the c variables that determined the class label. Feature values for all instances were chosen from a uniform distribution and the range of the number of instances was 100 to 2000.

The second dataset presented in this paper indicates that *Emca4*, a genetic determinant of susceptibility to 17β-estradiol (E2)-induced mammary cancer in the rat, has been mapped to rat chromosome 7 (RNO7) (Colletti et al., 2014; Schaffer et al., 2006; Shull, 2007). Data presented herein indicate that *Emca4* harbors multiple genetic determinants of mammary cancer susceptibility and tumor aggressiveness that are orthologous to breast cancer risk loci mapped to chromosome *8q24-24* in genome wide association studies (GWAS) (Easton et al., 2007; Turnbull et al., 2010; Fletcher et al., 2011; Michailidou et al., 2013; Ahsan et al., 2014; Michailidou et al., 2017).

The proceeding algorithm(s) used 76 of the SNPs in the designated region that are in the Hunter GWAS data set of 1145 breast cancer cases and 1142 controls (Hunter et al., 2007). All patients that had incomplete SNP data (630 patients) were omitted from our analysis. The data was made available via dbGaP's Cancer Genetic Markers of Susceptibility (CGEMS) Breast Cancer GWAS.

4.2. Synthetic Data Results

The following were the different CI functions investigated: parity (order two), order four, order five, and order six. For orders four, five, and six, ten different functions per order were chosen for analysis. To best represent the results, a learning curve was created to show the average AUC for n total features, where $n \in \{20, 50, 100\}$ (i.e. the average AUC of the ten different functions for each order, respectively). For each n, we trained datasets that contained m examples, where $m \in \{100, 200, 300, \ldots, 2000\}$. In addition, a learning curve was plotted to represent the average number of features that were kept using the same number of features and examples as stated above (i.e. the average number of features retained of the ten different functions for each order, respectively). To make the comparison fair, 10% percent of the total number of features remaining at a given iteration were removed. In (Guyon et al., 2002), the authors removed half of the features. However, we believe removing 10% of features at each iteration gave more accurate results since the number of features to begin with was not as large as the number in Guyon's paper.

In Figure 1 we show the synthetic data results for 20 total features across the CI functions of order two, four, five, and six. A learning curve was plotted to show the results of the average AUC and the average number of features retained across the various orders. The figures show the outcome for the RFEST and RFE algorithm. The performance (in terms of AUC) of both algorithms increased as the number of training examples increased, which is to be expected. However, RFEST achieved an AUC of 1.0 at a faster rate than RFE. In fact, for orders five and six, RFE failed to attain an AUC of 1.0.

The average number of features retained across the various orders was also calculated (second column of Figure 1). The goal was to output only the relevant features. For example, in the case of the parity function, we set the relevant features to be randomly chosen among the 20 features in our dataset, and the remaining features were irrelevant. For the CI function of order four, four randomly chosen features were set as the relevant variables, and the remaining features were irrelevant. The creation of the remaining CI functions followed a similar format. All feature values were chosen

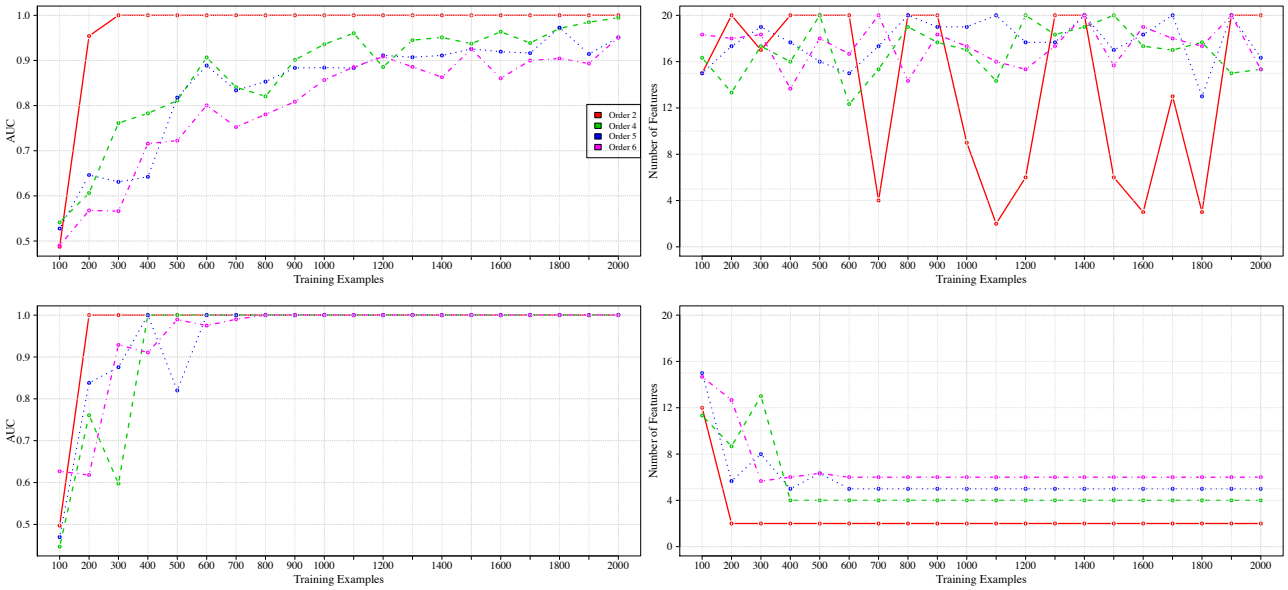

Figure 1. From left to right are results for 20 total features. The first column represents the AUC achieved across different training examples and the second column shows the number of features that were retained, with respect to the AUC achieved from the first column. The first row shows the results for RFE. The second row shows the results for RFEST.

with respect to a uniform distribution.

Observe that in Figure 1, RFE was not able to retain the relevant features across all orders. For orders four, five, and six, the algorithm stopped prematurely, outputting nearly all of the original features. In the case of the parity function, there are several instances where the RFE algorithm outputted only a small subset of features that included the relevant variables, however, it failed to return solely those that are relevant. Unlike RFE, RFEST was able to retain solely the relevant variables for each order. For the parity function, only 200 instances were required. For orders four, five, and six, 400, 400, and 300 instances were needed to return a subset of only the relevant features, respectively. This is a significant difference.

Figure 2 shows the synthetic data results for 50 total features across the CI functions of order two, four, five, and six. In a similar format to Figure 1, the first row represents the AUC achieved and features retained, respectively, for RFE. The second row represents the results for RFEST. As compared to Figure 1, there is a general decrease in AUC across all orders, as the number of irrelevant features increases. However, after a certain number of training examples, RFEST outperformed RFE. The max average AUC achieved for RFE and RFEST for all orders are represented in Table 2.

In addition to the significant difference in performance (as shown in Table 2), there is a distinct difference in the number of features returned. Across all orders and all varying training examples, RFE was not able to find the subset of relevant variables. However, RFEST was able to do so with the corresponding max average AUC's and training examples from Table 2. This observation is also shown graphically in the second column of Figure 2.

Table 2. Max average AUC results for 50 total features.

	RFE		RFEST	
ORDER	AUC	TRAINING EXAMPLES	AUC	TRAINING EXAMPLES
2	0.889	1600	1.0	400
4	0.773	1800	1.0	600
5	0.649	2000	1.0	900
6	0.710	2000	1.0	900

Lastly, Figure 3 shows the synthetic data results for 100 total features across the CI functions of order two, four, five, and six. Similar to the results in Figures 1 and 2, RFEST outperformed RFE in both prediction performance and the ability to retain fewer relevant features. The max average AUC results for RFE and RFEST can be found in Table 3. For 100 total features, at approximately 900 training examples, there is a significant difference between the prediction performance (across CI function orders two, four, and six) for RFE and RFEST (as shown in Table 3). That is, with fewer training examples, RFEST achieved higher max average AUCs compared to RFE.

Observe that in Figure 3, as the number of training examples increased, RFEST was able to return a smaller subset of features, whereas, RFE returned more than half the number of original features. Note that RFEST demonstrated more

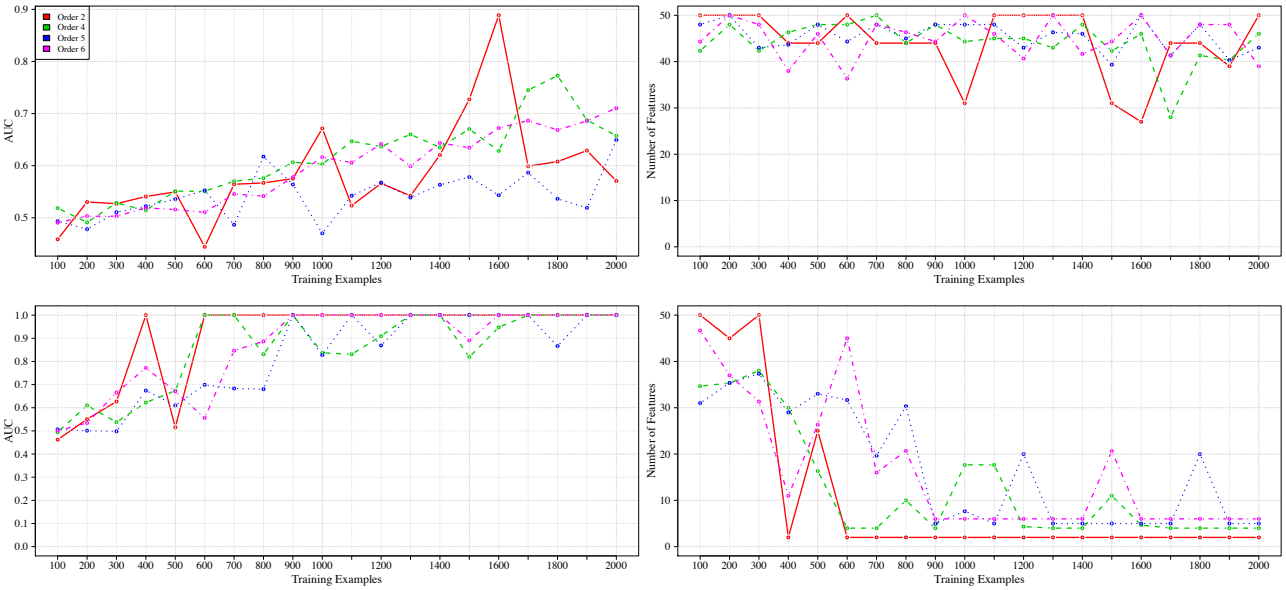

Figure 2. From left to right are results for 50 total features. The first column represents the AUC achieved across different training examples and the second column shows the number of features that were retained, with respect to the AUC achieved from the first column. The first row shows the results for RFE. The second row shows the results for RFEST.

fluctuation with 100 total features. This finding suggests that there are CI functions in which RFEST may not be the more robust method, in terms of its ability to return relevant features.

Table 3. Max average AUC results for 100 total features.

	RFE		RFEST	
ORDER	AUC	TRAINING EXAMPLES	AUC	TRAINING EXAMPLES
2	0.557	1100	1.0	900
4	0.624	1700	1.0	1600
5	0.548	1800	0.704	1500
6	0.611	1800	1.0	1400

4.3. Germline Genomic Data for Breast Cancer Results

There is great interest in associating variations in the human genome with disease risk. Much of this work focuses on associating with any given disease the variations in SNPs. Most such work assumes the SNPs, and the variations in disease risk that they cause, are independent of one another; in general this assumption is wrong and results in lost accuracy.

We may examine variations in the germline DNA with which a person is born or variations that arise from somatic mutations in individual cells, such as in the development of cancers and which may vary widely even within the same tumor. One particular disease for which both types of variations have been studied is breast cancer. For predicting disease risk, germline genomic data is the more natural choice to use.

The data we investigate contains 76 SNPs, translating to 152 binary features, in a particular region of the human genome that is orthologous to a region of the rat genome known to modulate breast cancer risk. We use the CGEMS data set of SNP genotypes for 1145 breast cancer cases and 1142 healthy age- and gender-matched controls (Hunter et al., 2007). Applying RFEST to this data, as run in the previous section, produces a cross-validated AUC of 0.56 with only nine features, which outperforms linear SVM and nonlinear SVM cross-validated runs with the original input data (0.53 and 0.54, respectively). Likewise, RFEST outperformed RFE as RFE (as run in the previous section) returned an AUC of 0.53 with 122 features, no better than either a linear or nonlinear SVM run. We also implemented RFE with the stopping criterion stated in Algorithm 1. That is, since RFEST returned nine features, RFE also iterated until nine features remained but returned an AUC of 0.51.

While all runs were performed by eliminating 10% of features at a time, our novel algorithm is also effective (when compared to RFE) when removing 20% or even 30% of the remaining features at a time. Removing 10% of the features at a time not only resulted in an AUC of 0.56 but most notably retained only nine features. With such a small set of features, one can then exhaustively generate all pairs (and even more) of interaction terms. A linear SVM model was built with the remaining nine features and all interaction terms. In turn, the top 13 features were all pairs of SNPs rather than individual SNPs. This suggests that interactions

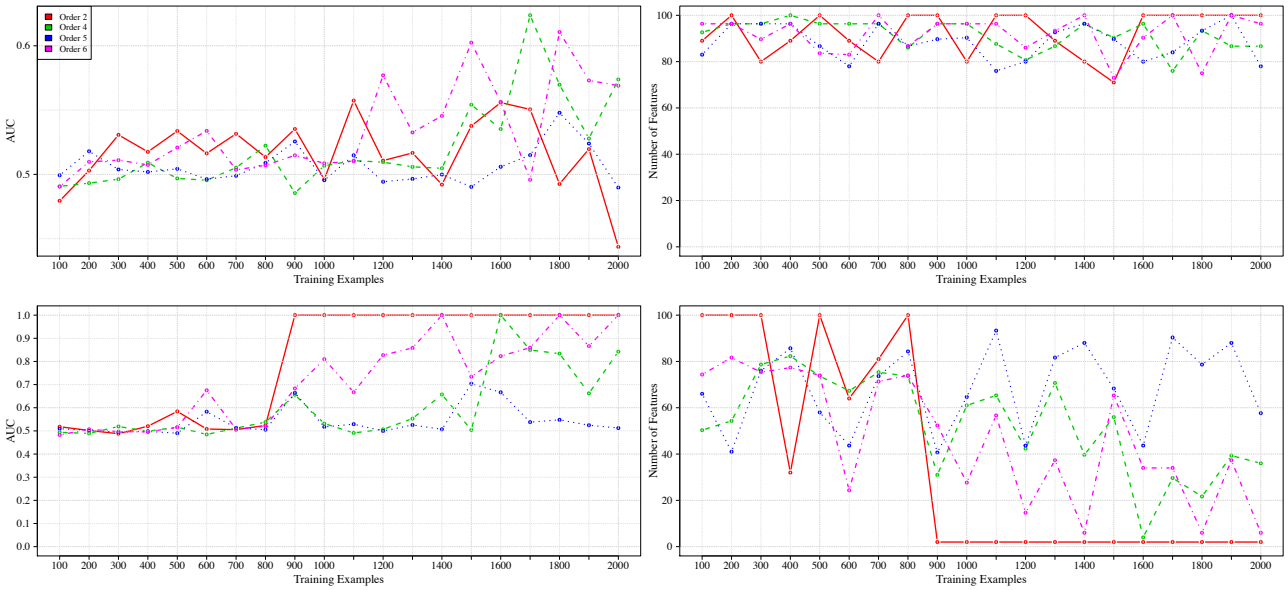

Figure 3. From left to right are results for 100 total features. The first column represents the AUC achieved across different training examples and the second column shows the number of features that were retained, with respect to the AUC achieved from the first column. The first row shows the results for RFE. The second row shows the results for RFEST.

play a major role in the effect of SNP variations in this region on breast cancer risk, as has been suspected. Studies are under way to further evaluate these nine selected SNPs.

It has been shown that incorporating, as risk factors, germline SNPs associated with breast cancer can significantly improve prediction and even mammography-based diagnosis of breast cancer, even though breast cancer is estimated to be only 30% heritable (Liu et al., 2014). In this section we show that avoiding the independence assumption regarding SNPs, by using a nonlinear SVM with our novel RFE algorithm, makes it possible to associate with breast cancer new SNPs and their interactions, and that this association can enable more accurate breast cancer risk prediction than could be made from these SNPs without taking interactions into account.

A post-hoc analysis of these nine selected SNPs confirmed our collaborating biologist's suspicion that interactions (rather than specific SNP values) were the most important modulator of breast cancer risk in this genomic region, and revealed which interactions were crucial to the underlying task (i.e. breast cancer diagnoses).

5. Discussion & Conclusion

In this paper, we explored the difficulties that accompany feature selection and learning decidedly nonlinear target concepts. In addition, we discussed the challenges that are faced in using Guyon's RFE algorithm (Guyon et al., 2002). Such problems occur in runtime for large datasets as well as a significantly lower AUC than the novel RFE algorithm.

We introduce a new algorithm, RFEST, for a nonlinear machine learning algorithm and demonstrate its efficacy both theoretically (refer to **Theorem 3.1** and Supplementary Material) and empirically (see **Section 4**). The RFE algorithm is an embedded-based approach but RFEST behaves like a wrapper-based approach. It uses a nonlinear SVM as a black box to determine feature relevance. In principle, with this approach one can use *any* machine learning algorithm to remove irrelevant or redundant features.

RFEST differs from RFE in two important ways: it perturbs rather than eliminates each feature to test sensitivity and measures loss in model efficacy instead of the loss in weighted sum of distances from the margin. These differences result in substantial improvements across CI functions and a real-world breast cancer genomics problem.

Extending the feature types used by RFEST is left for future work. Lastly, if one knew that the input data contained many correlated features, then applying a filter algorithm before RFEST will aid in removing redundant features.

Acknowledgements

References

Ahsan, H., Halpern, J., Kibriya, M.G., Pierce, B.L., Tong, L., Gamazon, E., . . . , and Whittemore, A.S. A genome-wide association study of early-onset breast cancer identifies pfkm as a novel breast cancer gene and supports a common genetic spectrum for breast cancer at any age. cancer epidemiology, biomarkers and prevention : a publication

of the american association for cancer research, cosponsored by the american society of preventive oncology. *Cancer Epidemiology and Prevention Biomarkers*, 23(4): 658–669, 2014.

Blum, A., Furst, M. L., Jackson, J. C., Kearns, M. J., Mansour, Y., and Rudich, S. Weakly learning DNF and characterizing statistical query learning using fourier analysis. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pp. 253–262, 1994. doi: 10.1145/195058.195147.

Breiman, L. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001. ISSN 0885-6125. doi: 10.1023/A:1010933404324.

Bshouty, N.H., Hancock, T.R., and Hellerstein, L. Learning boolean read-once formulas with arbitrary symmetric and constant fan-in gates. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pp. 1–15, New York, NY, USA, 1992. ACM. ISBN 0-89791-497-X. doi: 10.1145/130385.130386.

Caruana, R. and Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pp. 161–168, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143865.

Chandrashekar, G. and Sahin, F. A survey on feature selection methods. *Computers and Electrical Engineering*, 40(1):16 – 28, 2014. ISSN 0045-7906. doi: https://doi.org/10.1016/j.compeleceng.2013.11.024. 40th-year commemorative issue.

Cline, T. A male-specific lethal mutation in drosophila melanogaster that transforms sex. *Developmental Biology*, 72:266–275, 1979.

Colletti, J.A., Leland-Wavrin, K.M., Kurz, S.G., Hickman, M.P., Seiler, N.L., Samanas, N.B., . . . , and Shull, J.D. Validation of six genetic determinants of susceptibility to estrogen-induced mammary cancer in the rat and assessment of their relevance to breast cancer risk in humans. *G3*, 4(8):1385–1394, 2014.

Easton, D.F., Pooley, K.A., Dunning, A.M., Pharoah, P.D.P., Thompson, D., . . . , D.G. Ballingerand, and Ponder, B.A.J. Genome-wide association study identifies novel breast cancer susceptibility loci. *Nature*, 447(7148):1087–1093, 2007.

Fletcher, O., Johnson, N., Orr, N., Hosking, F.J., Gibson, L.J., Walker, K., . . . , and Peto, J. Novel breast cancer susceptibility locus at 9q31.2: Results of a genome-wide association study. *Journal of the National Cancer Institute*, 103:425–435, 2011.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, 2002.

Hsu, C., Chang, C., and Lin, C. A practical guide to support vector classification, 2010.

Hunter, D.J., Kraft, P., Jacobs, K.B., Cox, D.G., Yeager, M., Hankinson, S.E., . . . , and Chanock, S.J. A genome-wide association study identifies alleles in fgfr2 associated with risk of sporadic postmenopausal breast cancer. *Nat Genet*, 39(7):870–874, 2007.

Kampe, B., Kloß, S., Bocklitz, T., Rösch, P., and Popp, J. Recursive feature elimination in raman spectra with support vector machines. *Frontiers of Optoelectronics*, 10 (3):273–279, Sep 2017. ISSN 2095-2767. doi: 10.1007/s12200-017-0726-4.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004.

Kong, L., Kong, L., Wang, C., Jing, R., and Zhang, L. Predicting protein structural class for low-similarity sequences via novel evolutionary modes of pseaac and recursive feature elimination. *Letters in Organic Chemistry*, 14(9):673–683, 2017. ISSN 1570-1786/1875-6255. doi: 10.2174/1570178614666170511165837.

Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., . . . , and Hunt., T. *caret: Classification and Regression Training*, 2016. R package version 6.0-73.

Lal, TN., Chapelle, O., Weston, J., and Elisseeff, A. *Embedded methods*, pp. 137–165. Studies in Fuzziness and Soft Computing ; 207. Springer, Berlin, Germany, 2006.

Liu, J., Page, D., Peissig, P., McCarty, C., Onitilo, A.A., Trentham-Dietz, A., and Burnside, E. New genetic variants improve personalized breast cancer diagnosis. *AMIA Jt Summits Transl Sci Proc*, pp. 83–89, 2014.

Liu, Q., Chen, C., Zhang, Y., and Hu, Z. Feature selection for support vector machines with rbf kernel. *Artificial Intelligence Review*, 36(2):99–115, Aug 2011. ISSN 1573-7462. doi: 10.1007/s10462-011-9205-2.

Maldonado, S. and Weber, R. *Embedded Feature Selection for Support Vector Machines: State-of-the-Art and Future Challenges*. Springer Berlin Heidelberg, 2011.

Michailidou, K., Hall, P., Gonzalez-Neira, A., Ghoussaini, M., Dennis, J., Milne, R.L., . . . , and Easton, D.F. Large-scale genotyping identifies 41 new loci associated with breast cancer risk. *Nat Genet*, 45(4):353–361, 2013.

Michailidou, K., Lindstrm, S., Dennis, J., Beesley, J., Hui, S., Kar, S., ..., and Easton, D.F. Association analysis identifies 65 new breast cancer risk loci. *Nature*, 551:92 EP –, 2017.

Nguyen, M.H. and de la Torre, F. Optimal feature selection for support vector machines. *Pattern Recognition*, 43(3): 584 – 591, 2010. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2009.09.003.

Qureshi, M.N.I., Min, B., Jo, H.J., and Lee, B. Multiclass classification for the differential diagnosis on the adhd subtypes using recursive feature elimination and hierarchical extreme learning machine: Structural mri study. *PLOS ONE*, 11(8):1–20, 08 2016. doi: 10.1371/journal.pone.0160697.

Roy, B. A brief outline of research on correlation immune functions. In Batten, L. and Seberry, J. (eds.), *Information Security and Privacy*, pp. 379–394, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45450-2.

Schaffer, B.S., Lachel, C.M., Pennington, K.L., Murrin, C.R., Strecker, T.E., Tochacek, M., ..., and Shull, J.D. Genetic bases of estrogen-induced tumorigenesis in the rat: Mapping of loci controlling susceptibility to mammary cancer in a brown norway x aci intercross. *Cancer Research*, 66(15):7793–7800, 2006.

Schwartz, M.P., Hou, Z., Propson, N.E., Zhang, J., Engstrom, C.J., Costa, V.S., ..., and Thomson, J.A. Human pluripotent stem cell-derived neural constructs for predicting neural toxicity. *Proceedings of the National Academy of Sciences*, 112(40):12516–12521, 2015. doi: 10.1073/pnas.1516645112.

Shull, J.D. The rat oncogenome: Comparative genetics and genomics of rat models of mammary carcinogenesis. *Breast Disease*, 28(1):69–86, 2007.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. Rocr: visualizing classifier performance in r. *Bioinformatics*, 21(20):7881, 2005.

Stambaugh, C., Yang, H., and Breuer, F. Analytic feature selection for support vector machines. *CoRR*, abs/1304.5678, 2013.

Tao, P., Liu, T., Li, X., and Chen, L. Prediction of protein structural class using tri-gram probabilities of position-specific scoring matrix and recursive feature elimination. *Amino Acids*, 47(3):461–468, Mar 2015. ISSN 1438-2199. doi: 10.1007/s00726-014-1878-9.

Turnbull, C., Ahmed, S., Morrison, J., Pernet, D., Renwick, A., Maranian, M., ..., and Easton, D.F. Genome-wide association study identifies five new breast cancer susceptibility loci. *Nat Genet*, 42(6):504–507, 2010.

Wang, T., Huang, H., Tian, S., and Xu, J. Feature selection for svm via optimization of kernel polarization with gaussian ard kernels. *Expert Syst. Appl.*, 37(9): 6663–6668, September 2010. ISSN 0957-4174. doi: 10.1016/j.eswa.2010.03.054.

Wickham, H. The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, 40(1):1–29, 2011.

Zarogianni, E., Storkey, A.J., Johnstone, E.C., Owens, D.G.C., and Lawrie, S.M. Improved individualized prediction of schizophrenia in subjects at familial high risk, based on neuroanatomical data, schizotypal and neurocognitive features. *Schizophrenia Research*, 181:6 – 12, 2017. ISSN 0920-9964. doi: https://doi.org/10.1016/j.schres.2016.08.027.

Zhang, Z., Ersoz, E., Lai, C.Q., Todhunter, R.J., Tiwari, H.K., Gore, M.A., ..., and Buckler, E.S. Mixed linear model approach adapted for genome-wide association studies. *Nature*, 42:355 EP –, 2010.

Recursive Feature Elimination by Sensitivity Testing

Anonymous Authors[1]

1. Supplementary Material

1.1. Generalization of Theorem 2.1

In this Section, we will prove a stronger version of Theorem 2.1, generalizing it to apply to a product distribution $\mathcal{D}$ and to a function other than parity.

There are two parameters that are important in generalizing Theorem 2.1, ρ and $I_{\min}$. Under a uniform distribution, each feature j has equal probability of being either 1 or 0. Under a product distribution, one of these two probabilities may be larger than the other. We use $\rho > 0$ to denote the maximum, over all features j, of the ratio between the larger and the smaller of these two probabilities, for product distribution $\mathcal{D}$. Thus, for example, if each feature j is 1 with probability 3/4 and 0 with probability 1/4, then $\rho = 3$.

When the examples are labeled according to a parity function (on a subset of the variables), flipping the value of a relevant feature j in a random example drawn from $\mathcal{D}$ always changes the value of the function. For other functions g, flipping the value of a relevant feature j in a random example drawn from $\mathcal{D}$ will change the value of g with some non-zero probability. We denote the minimum of that probability, over all relevant j, by $I_{\min}$. This is the minimum *influence* of a relevant variable of g, with respect to distribution $\mathcal{D}$ (cf. (Hellerstein & Servedio, 2007)).

For the uniform distribution with g being a parity function, $\rho = 1$ and $I_{\min} = 1$.

The generalized theorem replaces the polynomial dependence of m on $\frac{1}{\frac{1}{2}-\epsilon}$ in Theorem 2.1 with a polynomial dependence on $\frac{1}{\frac{1}{2}I_{\min}-\rho\epsilon}$.

Theorem 1.1. *Suppose a machine learning algorithm is used to learn a classifier M for a Boolean target concept f defined on n Boolean features, where the target concept labels examples according to the value of a Boolean function g, computed on a fixed subset of the features. Suppose M has true error rate $\epsilon < \frac{1}{2}$, with respect to a product distribution $\mathcal{D}$, where $2\rho\epsilon \leq I_{\min}$. Then there is a quantity t that is polynomial in $n, \ln\frac{1}{\delta}, and \frac{1}{\frac{1}{2}I_{\min}-\rho\epsilon}$, with the following property: for all $0 < \delta < 1$, if the $\tilde{R}(j)$ values for all n features are computed using M and an i.i.d. sample of size t, drawn from distribution $\mathcal{D}$, then with probability least $1-\delta$, the computed $\tilde{R}(j)$ values for all the relevant features will be higher than the computed $\tilde{R}(j)$ values for the irrelevant features.*

Proof. Consider a random example a drawn from $\mathcal{D}$. Flipping any relevant bit in a reverses the output of f with probability at least $I_{\min}$.

Let $P(a)$ denote the probability of drawing assignment a from distribution $\mathcal{D}$. By the definition of ρ, for any bit j, $\frac{1}{\rho}P(a) \leq P(a_{\neg j}) \leq \rho P(a)$. Here $a_{\neg j}$ denotes the assignment produced by flipping bit j of a.

Let A denote the set of assignments in $\{0,1\}^n$ such that $M(a) \neq f(a)$.

Consider a relevant variable j of f. First, we will lower bound the probability, for random a drawn from distribution $\mathcal{D}$, that $f(a) \neq M(a_{\neg j})$. It is easy to see that $f(a) \neq M(a_{\neg j})$ iff one of the following two conditions holds: (1) $f(a) \neq f(a_{\neg j})$, and $a_{\neg j} \notin A$, or (2) $f(a) = f(a_{\neg j})$, and $a_{\neg j} \in A$. Thus the probability that $f(a) \neq M(a_{\neg j})$ is lower bounded by the probability that Condition (1) holds. We will now lower bound that probability.

$$\begin{aligned} &Prob[f(a) \neq f(a_{\neg j}) \text{ and } a_{\neg j} \notin A] \\ &\geq Prob[f(a) \neq f(a_{\neg j})] - Prob[a_{\neg j} \in A] \\ &\geq I_{\min} - \rho\epsilon \end{aligned} \tag{1}$$

The last inequality above uses the fact that the total probability mass of A is ϵ, and therefore the total probability mass of assignments a such that $a_{\neg j} \in A$ is at most $\rho\epsilon$.

Thus, for relevant variable j, for random a drawn from $\mathcal{D}$, $Prob[f(a) \neq M(a_{\neg j})] \geq I_{\min} - \rho\epsilon$.

Now consider the case where j is an irrelevant variable. In this case, the only way that $f(a) \neq M(a_{\neg j})$ is if $a_{\neg j} \in A$, which happens with probability at most $\rho\epsilon$. Therefore, $Prob[f(a) \neq M(a_{\neg j})] \leq \rho\epsilon$.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

In the statement of the theorem, we assumed that $I_{\min} > 2\rho\epsilon$. Let $\tau = \frac{1}{2}I_{\min} - \rho\epsilon$.

Now suppose we compute the $\tilde{R}(j)$ values for all features j using an i.i.d. random sample $\mathcal{X}$ drawn from $\mathcal{D}$ and labeled according to f. Let $t = \frac{1}{2\tau^2}\ln\frac{n}{\delta}$ be the size of this sample. Recall that $\tilde{R}(j)$ is the difference between the accuracy of M on $\mathcal{X}$, and the accuracy of M on the sample derived from $\mathcal{X}$ by flipping j in each example. This second accuracy measures the percentage of examples a for which $f(a) = M(a_{\neg j})$. Let $d(j)$ be the percentage of examples a for which $f(a) \neq M(a_{\neg j})$. It follows that for any pair of features j' and j'', $\tilde{R}(j') \geq \tilde{R}(j'')$ iff $d(j') \geq d(j'')$. We will prove the following claim: with probability at least $1-\delta$, $d(j) > \frac{1}{2}I_{min}$ for each relevant feature j, and $d(j) < \frac{1}{2}I_{min}$ for each irrelevant feature j. This suffices to prove the theorem.

To prove the claim, consider a random a drawn from $\mathcal{D}$. We can view the test of whether $f(a) \neq M(a_{\neg j})$ as a Bernoulli trial, with success when the inequality holds. Thus if j is a relevant variable, the probability of success is at least $I_{min} - \rho\epsilon$. If j is an irrelevant variable, the probability of success is at most $\rho\epsilon$.

With this view, we can apply a standard bound of Hoeffding. Consider a sequence of m independent Bernoulli trials, each with probability p of success. Suppose that out of these m trials, the observed fraction of successes is $\hat{p}$. The bound of Hoeffding states that for any $c > 0$, $Prob[\hat{p} \geq p + c] \leq e^{-2mc^2}$ (Hoeffding, 1963). By exchanging the role of failures and successes, it immediately follows that the inequality $Prob[\hat{p} \leq p - c] \leq e^{-2mc^2}$ also holds. Thus if $m \geq \frac{1}{2c^2}\ln\frac{1}{\delta}$, we have the following two inequalities

$$Prob[\hat{p} \geq p + t] \leq \delta \quad (2)$$

$$Prob[\hat{p} \leq p - t] \leq \delta \quad (3)$$

We apply these two inequalities to the tests performed in computing $d(j)$ from $\mathcal{X}$. Consider a random assignment a drawn from $\mathcal{D}$. If j is relevant, then the probability of success (i.e., that $f(a) \neq M(a_{\neg j})$) is at least $(I_{min} - \rho\epsilon)$. If j is irrelevant, then the probability of success is at most $\rho\epsilon$. The assignments in $\mathcal{X}$ correspond to $\frac{1}{2\tau^2}\ln\frac{n}{\delta}$ Bernoulli trials. Because $\tau = \frac{1}{2}I_{\min} - \rho\epsilon$, applying the above bounds with $c = \tau$ and $s = \frac{1}{2\tau^2}\ln\frac{n}{\delta}$ implies that the following holds for each feature j: If j is relevant, then $Prob[d(j) \leq \frac{1}{2}I_{\min}] \leq \frac{\delta}{n}$, and if j is irrelevant, then $Prob[d(j) \geq \frac{1}{2}I_{\min}] \leq \frac{\delta}{n}$.

Since there are n features, it follows that with probability at least $1-\delta$, the $d(j)$ values for the relevant variables will all be greater than $\frac{1}{2}I_{\min}$, and the $d(j)$ values for the irrelevant features will be less then $\frac{1}{2}I_{\min}$. □

The condition $\epsilon < I_{min}/(2\rho)$ in the above theorem limits its applicability to arbitrary functions g, even under the uniform distribution. For example, consider the consensus function (which is correlation immune): $g(x_1, \ldots, x_k) = 1$ iff $x_1 = x_2 = \ldots = x_k$. Under the uniform distribution, the value of I_{min} for the consensus function is $1/2^{k-2}$. For $k = 4$, the condition $\epsilon < I_{min}/(2\rho)$ would then be satisfied only if the error ϵ of model M was less than $1/8$.

We note that while it might be possible to prove a version of the theorem with a somewhat less restrictive condition, there are inherent limits as to what can be proved. For example, suppose g is a function on k variables that classifies at least 75% of its 2^k possible examples as negative. (The consensus function on 3 variables has this property.) Then the model that predicts negative on all examples has exactly 75% accuracy. Using RFEST with such a model, there is no hope of distinguishing relevant from irrelevant variables.

References

Hellerstein, L. and Servedio, R. A. On pac learning algorithms for rich boolean function classes. *Theoretical Computer Science*, 384(1):66 – 76, 2007. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2007.05.018. Theory and Applications of Models of Computation.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 01621459.