Math. Appl. **7** (2018), 11–30
DOI: 10.13164/ma.2018.02

RANDOM SUBSPACE LEARNING (RASSEL) WITH DATA DRIVEN WEIGHTING SCHEMES

MOHAMED ELSHRIF AND ERNEST FOKOUÉ

Abstract. We present a novel adaptation of the random subspace learning approach to regression analysis and classification of high dimension low sample size data, in which the use of the individual strength of each explanatory variable is harnessed to achieve a consistent selection of a predictively optimal collection of base learners. In the context of random subspace learning, random forest (RF) occupies a prominent place as can be seen by the vast number of extensions of the random forest idea and the multiplicity of machine learning applications of random forest. The adaptation of random subspace learning presented in this paper differs from random forest in the following ways: (a) instead of using trees as RF does, we use multiple linear regression (MLR) as our regression base learner and the generalized linear model (GLM) as our classification base learner and (b) rather than selecting the subset of variables uniformly as RF does, we present the new concept of sampling variables based on a multinomial distribution with weights (success 'probabilities') driven through p independent one-way analysis of variance (ANOVA) tests on the predictor variables. The proposed framework achieves two substantial benefits, namely, (1) the avoidance of the extra computational burden brought by the permutations needed by RF to de-correlate the predictor variables, and (2) the substantial reduction in the average test error gained with the base learners used.

1. Introduction

In machine learning, in order to improve the accuracy of a regression, or classification function, scholars tend to combine multiple estimators because it has been proven both theoretically and empirically [19, 20] that an appropriate combination of good base learners leads to a reduction in prediction error. This technique is known as ensemble learning (aggregation). In spite of the underlying algorithm used, the ensemble learning technique most of the time (on average) outperforms the single learning technique, especially for prediction purposes [21]. There are many approaches of performing ensemble learning. Among these, there are two popular ensemble learning techniques, bagging [3] and boosting [7]. Many variants of these two techniques have been studied previously such as random forest [4] and AdaBoost [8] and applied in a prediction. Our proposed method belongs to the subclass of ensemble learning methods known as random subspace learning. Given a dataset $\mathscr{D} = \{\mathbf{z}_i = (\mathbf{x}_i^\top, \mathrm{y}_i)^\top,\ i = 1, \cdots, n\}$, where

MSC (2010): primary 62H30, 62G08, 62H20, 68T10, 68T05.
Keywords: Random SubSpace Learning (RSSL), RASSEL.
The research was supported by RIT and QCRI (HBKU).

$\mathbf{x}_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})^\top \in \mathscr{X} \subset \mathbb{R}^p$ and $\mathrm{y}_i \in \mathscr{Y}$ are realizations of two random variables X and Y respectively, we seek to use the data $\mathscr{D}$ to build estimators $\widehat{f}$ of the underlying function f for predicting the response Y given the vector X of explanatory variables. In keeping with the standard in statistical learning theory, we will measure the predictive performance of any given function f using the theoretical risk functional given by

$$\mathcal{R}(f) = \mathbb{E}[\ell(Y, f(X))] = \int_{\mathscr{X}\times\mathscr{Y}} \ell(\mathbf{x}, \mathrm{y}) dP(\mathbf{x}, \mathrm{y}), \tag{1.1}$$

with the ideal scenario corresponding to the universally best function defined by

$$\widehat{f}^* = \arg\inf_f \{\mathcal{R}(f)\} = \arg\inf_f \{\mathbb{E}[\ell(Y, f(X))]\}. \tag{1.2}$$

For classification tasks, the default theoretical loss function is the zero-one loss $\ell(Y, f(X)) = 1_{\{Y \neq f(X)\}}$, for which the theoretical universal best defined in (1.2) is the Bayes classifier given by $f^*(\mathbf{x}) = \arg\max_{\mathrm{y}\in\mathscr{Y}} \{\Pr[Y = \mathrm{y}|\mathbf{x}]\}$. For regression tasks, the squared loss $\ell(Y, f(X)) = (Y - f(X))^2$ is by far the most commonly used, mainly because of the wide variety of statistical, mathematical and computational benefits it offers. For regression under the squared loss, the universal best defined in (1.2) is also known theoretically to be the conditional expectation of Y given X, specifically given by $f^*(\mathbf{x}) = \mathbb{E}[Y|X = \mathbf{x}]$. Unfortunately, the aforementioned expressions of the best estimators cannot be realized in practice because the distribution function $P(\mathbf{x}, \mathrm{y})$ of (X, Y) defined on $\mathscr{X} \times \mathscr{Y}$ is unknown. To circumvent this learning challenge, one has to do essentially two foundational things, namely: (a) choose a certain function class $\mathscr{F}$ (approximation) from which to search for the estimator $\widehat{f}$ of the true but unknown underlying f, (b) specify the empirical version of (1.1) based on the given sample $\mathscr{D}$, an use that empirical risk as the practical objective function. However, in this paper, we do not directly construct our estimating classification functions from the empirical risk. Instead, we build the estimators using other optimality criteria, and then compare their predictive performances using the average test error $\mathtt{AVTE}(\cdot)$, namely

$$\mathtt{AVTE}(\widehat{f}) = \frac{1}{R}\sum_{r=1}^{R}\left\{\frac{1}{m}\sum_{t=1}^{m} \ell(\mathbf{y}_{i_t}^{(r)}, \widehat{f}_r(\mathbf{x}_{i_t}^{(r)}))\right\},$$

where $\widehat{f}_r(\cdot)$ is the r-th realization of the estimator $\widehat{f}(\cdot)$ built using the training portion of the split of $\mathscr{D}$ into training set and test set, and $\left(\mathbf{x}_{i_t}^{(r)}, \mathbf{y}_{i_t}^{(r)}\right)$ is the t-th observation from the test set at the r-th random replication of the split of $\mathscr{D}$. In this paper, we consider both multiclass classification tasks with response space $\mathscr{Y} = \{1, 2, \cdots, G\}$ and regression tasks with $\mathscr{Y} = \mathbb{R}$, and we focus on learning machines from a function class $\mathscr{F}$ whose members are ensemble learners.

Bootstrap Aggregating also known as bagging [3], boosting [9], random forests [4], and bagging with subspaces [18] are all predictive learning methods based on the ensemble learning principle for which the ensemble is built from the provided dataset $\mathscr{D}$ and the weights are typically taken to be equal.

In this paper, we focus on learning tasks involving high dimension low sample size (HDLSS) data, and we further zero-in on those datasets for which the number of explanatory variables p is substantially larger than the sample size n. As our main contribution in this paper, we introduce, develop, and apply a new adaptation of the theme of random subspace learning [14] using the traditional multiple linear regression (MLR) model as our base learner in regression and the generalized linear model (GLM) as a base learner in classification. Some applications by nature posses few instances (small n) with large number of features ($p \ggg n$) such as fMRI [16] and DNA microarrays [2] datasets. It is hard for a traditional algorithm to build a regression model, or to classify the dataset when it possesses a very small instances to features ratio. The prediction problem becomes even more difficult when this huge number of features are highly correlated, or irrelevant for the task of building such a model, as we will show later in this paper. Therefore, we harness the power of our proposed adaptive subspace learning technique to guide the choice/selection of good candidate features from the dataset, and therefore select the best base learners, and ultimately the ensemble yielding the lowest possible prediction error. In most typical random subspace learning algorithms, the features are selected according to an *equally likely* scheme. The question then arises as to whether one can *devise a better scheme to choose the candidate features for efficiently with some predictive benefits*. On the other hand, it is interesting to assess *the accuracy of our proposed algorithm under different levels of the correlation of the features*. The answer to this question constitutes one of the central aspect of our proposed method, in the sense *we explore a variety of weighting schemes for choosing the features, most of them (the schemes) based on statistical measures of relationship between the response variable and each explanatory variable*. As the computational section will reveal, the weighting schemes proposed here lead to an improvement in predictive performance of our method over random forest on most tested datasets because our framework leverages the accuracy of the learning algorithm through selecting many good models (*since the weighting scheme allows good variables to be selected more often which leads to near optimal base learners*).

2. Related work

Traditionally, in a prediction problem, a single model is built based on the training set and the prediction is decided based solely on this single fitted model. However, in bagging, bootstrap samples are taken from the dataset, then, for each instance, the model is fitted. Finally, the prediction is made based on the average of all bagged models. Mathematically, the prediction accuracy for the constructed model using bagging outperforms the traditional model and in the worst case it has the same performance. However, it must be said that it depends on the stability of the modeling procedure. It turns out that bagging reduces the variance without affecting the bias, thereby leading to an overall reduction in prediction error, and hence its great appeal. Any set of predictive models can be used as an ensemble in the sense defined earlier. There are many ensemble learning approaches. These approaches could be categorized into four classes: (1) algorithms that use heterogeneous predictive models such as stacking [22]. (2) algorithms that manipulate the instances of the datasets such as bagging [3], boosting [9], random forests [4], and bagging with subspaces [18]. (3) algorithms that maniplulate the features of the datasets such as random forests [4], random subspaces [14], and bagging with subspaces [18]. (4) algorithms that manipulate the learning algorithm such as random forests [4], neural networks ensemble [13], and extra-trees ensemble [10]. Since our proposed algorithm manipulates both the instances and features of the datasets, we will focus on the algorithms in the second and third categories [3, 4, 14, 18].

Bagging [3], or bootstrap aggregating is an ensemble learning method that generates multiple predictive models. These models are based on performing bootstrap replicates of the learning (training) dataset and utilizing from each replicate to build a separate predictive model. The bootstrap sample is attained through randomly (uniformly) sampling with replacement from instances of the training dataset. The decision is made based on averaging the predictor classifiers in regression task and taking the majority vote in classification task. Bagging tend to decrease the variance and keeps the bias as in the case of a single classifier. The bagging accuracy increases when the applied learner is unstable, which means that for any small fluctuation on the training dataset causes large impact on the test dataset such as trees [3].

Random forests [4], is an ensemble learning method that averages the prediction results from multiple independent predictor (tree) models. It also performs bootstrap replicates, like bagging [3], to construct different predictors. For each node of the tree, randomly selecting subset of the attributes. It is considered to improve over bagging through *de-correlating the trees. Choose the best attribute from the selected subset.* As [5] mentions that when building a random tree, there are three issues that should be decided in advance; (1) the leafs splitting method, (2) the type of predictor, and (3) the randomness method.

Random subspace learning [14], is an ensemble learning method that constructs base models based on different features. It chooses a subset of features and then learns the base model depending only on these features. The random subspaces reaches the *highest* accuracy when the number of features is large as well as the number of instances. In addition, it performs *good* when there are redundant features on the dataset.

Bagging subspaces [18], is an ensemble learning method that combines both the bagging [3] and random subspaces [14] learning methods. It generates a bootstrap replicates of the training dataset, in the same way as bagging. Then, it randomly chooses a subset from the features, in the same manner as random subspaces. It outperforms the bagging and random subspaces. Also, it is found to yield the same performance as random forests in case of using decision tree as a base learner.

In the simulation part of this paper, we aim to answer the following research questions: (1) Is the performance of the adaptive random subspace learning (RSSL) better than the performance of single classifiers? (2) What is the performance of the adaptive RSSL compared to the most widely used classifier ensembles? (3) Is there a theoretical explanation as to why adaptive RSSL works well for most of the simulated and real-life datasets? (4) How does adaptive RSSL perform on different parameter settings and with various percentages of the instance-to-feature ratio (IFR)? (5) How does the correlation between features affects the prediction performance of the adaptive RSSL algorithm?

3. Problem formulation

Thanks to its tremendous success in achieving accurate predictions on a wide variety of high dimensional datasets, the random forest algorithm has been the subject of many recent studies, with many authors adapting the main idea or modifying various aspects of how the base learners are constructed to further achieve greater predictive performances. [11] for instance presents a recent application of random forest in genomics with some modification on the original methodological used for building the random forest ensemble. [6] used random forest for both gene selection and gene classification in DNA Microarray data, benefitting from the variable importance measure that is offered as a byproduct of random forest. In recent years, many studies have explored extensions and improvement of the original random forest idea, some in a spirit similar to our present work like [23] who select the features for the subspace using weights inspired by the relationship between a given variable and the response. In the context of high dimensional response (output) space, [15] is yet another interesting adaptation of random forest aimed at attaining every greater predictive performances. [1] and [17] have also recently proposed very interesting extension on the RF theme that seeks to further lower the prediction error. Coming from a perspective similar to ours, even though their base learners are still trees whereas we allow any type of base learner, [1] enriches the RF ensemble by way of a random subspace selection that gives less weight to weak features, i.e. features with weaker relationship with the response.

As stated earlier, our proposed method belongs to the category of random subspace learning where each base learner is constructed using a bootstrap sample and a subset of the original p features. The main difference here is that we use base learners that are typically considered not to lead to any improvement when aggregated. [4] and [3] for instance clearly states that linear discriminant analyzers for instance cannot benefit from being bagged.

Some authors before use, like [23], in their recent work stratified sampling for feature subspace selection in random forests for high dimensional data, have weighted the trees comprising the random subspace ensembles. However, the manner in which they implement data-driven weights in their stratified sampling scheme is markedly different from our method. First and foremost, our proposed approach is straightforward, intuitively appealing, easy to implement and computational very efficient compared to the other approaches. One key aspect of our method lies in the fact that we select features using data-driven weighting schemes that are functions of the individual strength of the relationship between each feature and the response (target). In fact, we conjecture that the way we construct the subspace indirectly de-correlates the base learners and thereby contributes to the substantial reduction in prediction error. Each base learner is driven by the subset $\{j_1^{(l)}, \cdots, j_d^{(l)}\} \subset \{1, 2, \cdots, p\}$ of d variables of predictors that are randomly selected according to a multinomial distribution to build it, and the subsample $\mathscr{D}^{(l)}$ drawn with replacement from $\mathscr{D}$. Our proposed algorithm 1 code-named RASSEL (Random Adaptive Subspace Ensemble Learner), in the spirit of random forest and all other random subspace learning methods, consists of building an ensemble of L base learners herein denoted $\mathcal{G}_{\texttt{RASSEL}} = \{\widehat{g}^{(1)}, \cdots, \widehat{g}^{(l)}, \cdots, \widehat{g}^{(L)}\}$, and forming

Algorithm 1 Building an ensemble of L base learners with using RASSEL

```
Given $\mathscr{D} = \{\mathbf{z}_i = (\mathbf{x}_i^\top, \mathrm{y}_i)^\top,\ i = 1, \cdots, n\}$, with $\mathbf{x}_i^\top = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})$ and $\mathrm{y}_i \in \mathscr{Y}$
Choose a base learner $g(\cdot)$
Choose an estimation/learning method
Construct the weighting scheme $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_p)$ from the data
for $l = 1$ to $L$ do
    Draw with replacement $\mathscr{D}^{(l)} = \{\mathbf{z}_1^{(l)}, \cdots, \mathbf{z}_n^{(l)}\}$ from $\mathscr{D}$
    Draw without replacement from $\{1, \cdots, p\}$ a subset $\mathcal{V}^{(l)} = \{j_1^{(l)}, \cdots, j_q^{(l)}\}$ of $q$ variables according
    to a multinomial distribution with success probabilities (weighting scheme) $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_p)$.
    Form the indicator vector $\boldsymbol{\gamma}^{(l)} = (\gamma_j^{(l)}, \cdots, \gamma_p^{(l)})$ with
        $\gamma_j^{(l)} = \begin{cases} 1 & \text{if } j \in \{j_1^{(l)}, \cdots, j_q^{(l)}\} \\ 0 & \text{otherwise.} \end{cases}$    (3.1)
    Drop from $\mathscr{D}^{(l)}$ all $j \notin \mathcal{V}^{(l)}$ and form $\mathscr{D}^{(l)}(\mathcal{V}^{(l)}) = \mathscr{D}^{(l)}(\boldsymbol{\gamma}^{(l)})$
    Build the $l$th base learner $\widehat{g}(\cdot, \boldsymbol{\gamma}^{(l)}) = \widehat{g}^{(l)}$ based on $\mathscr{D}^{(l)}(\mathcal{V}^{(l)})$
end for
Form the RASSEL ensemble $\mathcal{G}_{\texttt{RASSEL}}$
        $\mathcal{G}_{\texttt{RASSEL}} = \{\widehat{g}^{(1)}, \cdots, \widehat{g}^{(l)}, \cdots, \widehat{g}^{(L)}\}$
The ensemble prediction function is given by
        $\widehat{f}^{(L)}(\cdot) = \frac{1}{L}\sum_{l=1}^{L} \widehat{g}^{(l)}(\cdot)$
```

the ensemble prediction function as

$$\widehat{f}^{(L)}(\cdot) = \frac{1}{L}\sum_{l=1}^{L} \widehat{g}^{(l)}(\cdot). \tag{3.2}$$

Using the above ensemble $\mathcal{G}_{\texttt{RASSEL}} = \left\{\widehat{g}^{(l)}(\cdot),\ l = 1, \cdots, L\right\}$, we perform classification by predicting the class membership of $\mathbf{x}^* \in \mathscr{X}$ using the ensemble predicting estimator

$$\widehat{f}^{(L)}(\mathbf{x}^*) = \arg\max_{\mathrm{y} \in \mathscr{Y}} \left\{ \sum_{l=1}^{L} \left(\mathbf{1}_{\{\mathrm{y} = \widehat{g}^{(l)}(\mathbf{x}^*)\}} \right) \right\}.$$

For regression tasks, given $\mathbf{x}^* \in \mathscr{X}$, we predict its corresponding response using

$$\widehat{f}^{(L)}(\mathbf{x}^*) = \frac{1}{L}\sum_{l=1}^{L} \hat{g}^{(l)}(\mathbf{x}^*).$$

After formulating the objective functions of our proposed framework, the next step is to choose the weighting scheme and extract the dimension of the RASSEL algorithm.

4. Data-driven weighting scheme for subspace construction

One of the most typical and most obvious ingredients in the above proposed algorithm is the dimension d of the subspace. This is crucial, because its value has a strong bearing on an important aspect of the ensemble, namely the correlation among the base learners. [4] recommends using $d = \lceil\sqrt{p}\rceil$ in classification and $d = \lceil p/3 \rceil$ in regression, which both are reasonable in the $n \ggg p$ setting. In the $n \lll p$ context, we find the following to be reasonable: (a) for classification

Algorithm 2 Extracting Important variables for regression

```
Set $d \lll p$, the number of variables to be drawn
for $j = 1, \cdots, p$ do
    Compute $r_j = r(\mathtt{x}_j, \mathtt{y}) = \mathtt{correlation}(\mathtt{x}_j, \mathtt{y})$
end for
Create the vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_p)^\top$ where
    $$\pi_j = r_j^2 / \sum_{j'=1}^{p} r_{j'}^2$$
so that $0 < \pi_j < 1$ and $\sum_{j=1}^{p} \pi_j = 1$.
for $k = 1, \cdots, d$ do
    Draw without replacement from a multinomial with probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_p)^\top$,
    specifically
    $j_k \sim \mathtt{Multinomial}(\pi_1, \pi_2, \cdots, \pi_p)$
end for
Use the variables with drawn indices $\{j_1, j_2, \cdots, j_q\} \subset \{1, 2, \cdots, p\}$ as the basis of your
subspace.
```

$d = \min\left(\lceil n/5 \rceil, \lceil \sqrt{p} \rceil\right)$; (b) for regression $d = \min\left(\lceil n/5 \rceil, \lceil p/3 \rceil\right)$. See also the work of [12]. It turns out that the ensemble prediction function of (3.2) has variance $\mathbb{V}\left(\widehat{f}^{(L)}(\cdot)\right) = \sigma^2\psi + \frac{(1-\psi)}{L}\sigma^2$, where L is the number of base learners, and for $l \neq l'$ with $l, l' = 1, 2, \cdots, L$, $\psi = \mathtt{correlation}\left(\widehat{g}^{(l)}(\cdot), \widehat{g}^{(l')}(\cdot)\right)$ represents the correlation between two base learners, and for $l = 1, \cdots, L$, $\sigma^2 = \mathtt{var}\left(\widehat{g}^{(l)}(\cdot)\right)$. It turns out that by using our weighting scheme that favors individually strong variables, we end up reducing the correlation between base learners substantially, which we conjecture helps achieve the competitive predictive performances we noticed.

This is crucial step because it helps in selecting the best base learners. The first and arguably most naturally way to weight variables in a regression analysis context is their individual coefficient of determination. Given $\mathtt{x}_{1j}, \mathtt{x}_{2j}, \cdots, \mathtt{x}_{nj}$ and $\mathtt{y}_1, \mathtt{y}_2, \cdots, \mathtt{y}_n$, the so-called Pearson sample correlation coefficient is given by

$$r_j = r(\mathtt{x}_j, \mathtt{y}) = \frac{1}{n-1} \sum_{l=1}^{n} \left(\frac{\mathtt{x}_{lj} - \bar{\mathtt{x}}_j}{s_{\mathtt{x}_j}} \right) \left(\frac{\mathtt{y}_\ell - \bar{\mathtt{y}}}{s_{\mathtt{y}}} \right).$$

For the univariate regression $Y_i = \beta_0 + \beta_1 \mathtt{x}_{ij} + \epsilon_i$, it is known that $r_j^2 = R_j^2$, *the coefficient of determination*, measures the percentage of variation in Y that is explained (captured) by X_j through the regression line. Therefore, given p variables $X_1, X_2, \cdots, X_p$ in a linear regression setting/context, the variable X_j with the largest r_j^2 should be preferred over the others if we had to choose exactly one variable. If instead we have to choose more than one variable out of the p available, it makes sense to assign higher weights according to the individual r_j^2 of each X_j so as to give more important variables a greater chance of being chosen. $r_j^2 = R_j^2$ is a good measure of individual importance (at least prior importance) of X_j and can be used in random subspace learning as explained in algorithm 2. Instead of using R_j^2, one could consider using the corresponding F statistic $F_j = \frac{(n-2)r_j^2}{1-r_j^2}$.

Algorithm 3 Extracting Important variables for classification

```
Set $d \lll p$, the number of variables to be drawn
for $j = 1, \cdots, p$ do
    Compute the ANOVA F-statistic
end for
    $$F_j = \frac{\texttt{Between groups mean squared by x}_j}{\texttt{Within groups mean squared by x}_j}$$
Create the vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_p)^\top$ where
    $$\pi_j = F_j / \sum_{j'=1}^{p} F_{j'}$$
for $k = 1, \cdots, q$ do
    Draw without replacement from a multinomial with probabilities $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_p)^\top$, specifically
    $j_k \sim$ Multinomial$(\pi_1, \pi_2, \cdots, \pi_p)$
end for
Use the variables with drawn indices $\{j_1, j_2, \cdots, j_q\} \subset \{1, 2, \cdots, p\}$ as the basis of your subspace.
```

For classification tasks under the assumption that $X_j \sim N(\mu_{jy}, \sigma_j^2)$, with factor levels $y \in \{1, 2, \cdots, G\}$, algorithm 3 explains in detail how to extract the important features.

5. Adaptive RASSEL for MLR and GLM

Base Learners for Regression: Linear Model

Given $\{(\mathbf{x}_i, \mathrm{y}_i),\ i = 1, \cdots, n\}$, where $\mathbf{x}_i^\top = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})$ and $\mathrm{y}_i \in \mathbb{R}$. Assume the multiple linear regression (MLR) model

$$Y_i = \beta_0 + \beta_1 \mathrm{x}_{i1} + \beta_2 \mathrm{x}_{i2} + \cdots + \beta_p \mathrm{x}_{ip} + \epsilon_i. \tag{5.1}$$

If Equation (5.1) is applied to the whole training set, and using

$$\boldsymbol{\beta}^\top = (\beta_0, \beta_1, \beta_2, \cdots, \beta_p), \quad \boldsymbol{Y}^\top = (Y_1, Y_2, \cdots, Y_n) \quad \text{and} \quad \boldsymbol{\epsilon}^\top = (\epsilon_1, \epsilon_2, \cdots, \epsilon_n),$$

we can write

$$\boldsymbol{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{X} \in \mathbb{R}^{n \times (p+1)}$ is the design matrix defined by

$$\mathbf{X} = \begin{bmatrix} 1 & \mathrm{x}_{11} & \mathrm{x}_{12} & \cdots & \mathrm{x}_{1j} & \cdots & \mathrm{x}_{1p} \\ 1 & \mathrm{x}_{21} & \mathrm{x}_{22} & \cdots & \mathrm{x}_{2j} & \cdots & \mathrm{x}_{2p} \\ \vdots & \vdots & \vdots & \cdots & \ddots & \cdots & \vdots \\ 1 & \mathrm{x}_{i1} & \mathrm{x}_{i2} & \cdots & \mathrm{x}_{ij} & \cdots & \mathrm{x}_{ip} \\ \vdots & \vdots & \vdots & \cdots & \ddots & \cdots & \vdots \\ 1 & \mathrm{x}_{n1} & \mathrm{x}_{n2} & \cdots & \mathrm{x}_{nj} & \cdots & \mathrm{x}_{np} \end{bmatrix}.$$

It is a basic result in regression analysis theory that the ordinary least squares (OLS) estimator $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is the minimizer of $SSE(\boldsymbol{\beta})$, namely

$$\hat{\boldsymbol{\beta}}^{(\texttt{OLS})} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \{SSE(\boldsymbol{\beta})\} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \left\{(\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta})^\top (\boldsymbol{Y} - \mathbf{X}\boldsymbol{\beta})\right\},$$

which turns out to be the ubiquitous

$$\hat{\boldsymbol{\beta}}^{(\mathtt{OLS})} = \hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{Y}.$$

Given a new point $\mathbf{x}^* = (\mathrm{x}_1^*, \cdots, \mathrm{x}_p^*)^\top$ for which an predicted response value is desired, form $\tilde{\mathbf{x}}^{*\top} = (1, \mathrm{x}_1^*, \mathrm{x}_2^*, \cdots, \mathrm{x}_p^*)$, then simply compute

$$\widehat{Y}^*_{\mathtt{mlr}} = \widehat{g}_{\mathtt{mlr}}(\mathbf{x}^*) = \sum_{j=0}^{p} \hat{\beta}_j \mathrm{x}_j^* = \hat{\boldsymbol{\beta}}^\top \tilde{\mathbf{x}}^*. \tag{5.2}$$

Note that if $n \lll p$, then the prediction in (5.2) cannot be computed, because of the singularity of $\mathbf{X}^\top\mathbf{X}$.

Base Learners for Classification: Logistic Regression

Given $\mathbf{x}_i = (\mathrm{x}_{i1}, \cdots, \mathrm{x}_{ip})^\top \in \mathcal{X} \subseteq \mathbb{R}^p$, $Y_i \in \{0, 1\}$, and dataset

$$\mathcal{D} = \Big\{(\boldsymbol{x}_1, Y_1), \cdots, (\boldsymbol{x}_n, Y_n)\Big\}$$

Logistic Regression assumes that the response variable Y_i is related to the explanatory vector $\mathbf{x}_i$ through the model,

$$\log\left[\frac{\pi_i}{1-\pi_i}\right] = \eta(\mathbf{x}_i; \boldsymbol{\beta}) = \tilde{\mathbf{x}}_i^\top \boldsymbol{\beta}$$

where $\tilde{\mathbf{x}}_i = (1, \mathrm{x}_{i1}, \mathrm{x}_{i2}, \cdots, \mathrm{x}_{ip})^\top$, $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_p)^\top$,

$$\eta(\mathbf{x}_i; \boldsymbol{\beta}) = \tilde{\mathbf{x}}_i^\top \boldsymbol{\beta} = \beta_0 + \beta_1 \mathrm{x}_{i1} + \beta_2 \mathrm{x}_{i2} + \cdots + \beta_p \mathrm{x}_{ip}$$

and

$$\pi_i = \Pr[Y_i = 1 | \mathbf{x}_i, \boldsymbol{\beta}] = \frac{e^{\eta(\mathbf{x}_i;\boldsymbol{\beta})}}{1 + e^{\eta(\mathbf{x}_i;\boldsymbol{\beta})}} = \frac{1}{1 + e^{-\eta(\mathbf{x}_i;\boldsymbol{\beta})}} = \pi(\mathbf{x}_i; \boldsymbol{\beta}).$$

Base Learners for Classification: Logistic Regression Majority rule

Given a new vector $\mathbf{x}$,

$$Y_{\mathtt{glm}} = g_{\mathtt{glm}}(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad \Pr[Y = 1|\mathbf{x}, \boldsymbol{\beta}] = h(\tilde{\mathbf{x}}^\top \boldsymbol{\beta}) > \frac{1}{2}, \\ 0 & \text{if} \quad \text{otherwise.} \end{cases}$$

In other words, assign $\mathbf{x}$ to the class with the highest probability. The corresponding decision boundary is the set

$$\Big\{\mathbf{x} \in \mathcal{X} : \; h(\tilde{\mathbf{x}}^\top \boldsymbol{\beta}) - \frac{1}{2} = 0\Big\}.$$

Adaptive threshold: In some applications, the experimenter/researcher/data scientist may choose to use a threshold other than 1/2. If $\tau \neq 1/2$ is such a cutoff, then the decision boundary simply becomes

$$\Big\{\mathbf{x} \in \mathcal{X} : \; h(\tilde{\mathbf{x}}^\top \boldsymbol{\beta}) - \tau = 0\Big\}.$$

Base Learners for Classification: Logistic Regression

Let $\ell(\boldsymbol{\beta}) = \log L(\mathbf{y};\boldsymbol{\beta})$ denote the log-likelihood for the logistic regression model, then

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \mathrm{y}_i \tilde{\mathbf{x}}_i^\top \boldsymbol{\beta} - \sum_{i=1}^{n} \log[1 + \exp(\tilde{\mathbf{x}}_i^\top \boldsymbol{\beta})].$$

The maximum likelihood estimator of of $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^p} \Big\{ \ell(\boldsymbol{\beta}) \Big\}.$$

The estimator of the logistic regression base learner is then

$$\widehat{Y}^*_{\texttt{glm}} = \widehat{g}_{\texttt{glm}}(\mathbf{x}^*) = \arg\max_{j\in\mathcal{Y}} \Big\{ \Pr[Y = j|\mathbf{x}^*, \widehat{\beta}] \Big\}.$$

Multiclass Logistic Regression

Let $\mathcal{Y} = \{1, 2, \cdots, G\}$ be the set of class labels. Let $\boldsymbol{\beta}_j = (\beta_{j0}, \beta_{j1}, \cdots, \beta_{jp})^\top$ represent the vector of regression coefficients in group $j \in 1, 2, \cdots, G-1$ Set $\eta_G(\mathbf{x}) = 0$ and for $j = 1, 2, \cdots, G-1$, define

$$\eta_j(\mathbf{x}) = \beta_{j0} + \beta_{j1}\mathrm{x}_1 + \cdots + \beta_{jp}\mathrm{x}_p = \tilde{\mathbf{x}}^\top \boldsymbol{\beta}_j.$$

Define

$$\Pr[Y = j|\mathbf{x}] = \frac{e^{\eta_j(\mathbf{x})}}{1 + \displaystyle\sum_{l=1}^{G-1} e^{\eta_l(\mathbf{x})}}$$

and

$$\Pr[Y = G|\mathbf{x}] = \frac{1}{1 + \displaystyle\sum_{l=1}^{G-1} e^{\eta_l(\mathbf{x})}}.$$

We can then write

$$\log\left[\frac{\Pr[Y = j|\mathbf{x}]}{\Pr[Y = 0|\mathbf{x}]}\right] = \eta_j(\mathbf{x}) = \beta_{j0} + \beta_{j1}\mathrm{x}_1 + \cdots + \beta_{jp}\mathrm{x}_p = \tilde{\mathbf{x}}^\top \boldsymbol{\beta}_j.$$

6. Theoretical justification

We now consider an ensemble of L base learners, $\widehat{f}^{(l)}$ for $l = 1, 2, \cdots, L$, where $\widehat{f}^{(l)} = \widehat{\beta}_0 + \mathbf{x}^\top \widehat{\boldsymbol{\beta}}$. Each $\widehat{f}^{(l)}(\cdot)$ is built on a bootstrap sample from $\mathscr{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, ..., n\}$

$$\widehat{Y}_* = \frac{1}{L}(\widehat{f}^{(1)}(\mathbf{x}_*) + \widehat{f}^{(2)}(\mathbf{x}_*) + \cdots + \widehat{f}^{(L)}(\mathbf{x}_*)).$$

Proof. For each bootstrap sample l, the corresponding multiple linear regression (MLR) base learner $g^{(l)}(\cdot)$ predicts the response for a given $\mathbf{x}$ as

$$\widehat{g}^{(l)}(\mathbf{x}) = \mathbf{x}^\top \widehat{\boldsymbol{\beta}}^{(l)},$$

where

$$\widehat{\boldsymbol{\beta}}^{(l)} = ((\boldsymbol{X}^{(l)})^\top \boldsymbol{X}^{(l)})^{-1} (\boldsymbol{X}^{(l)})^\top \boldsymbol{Y}^{(l)}.$$

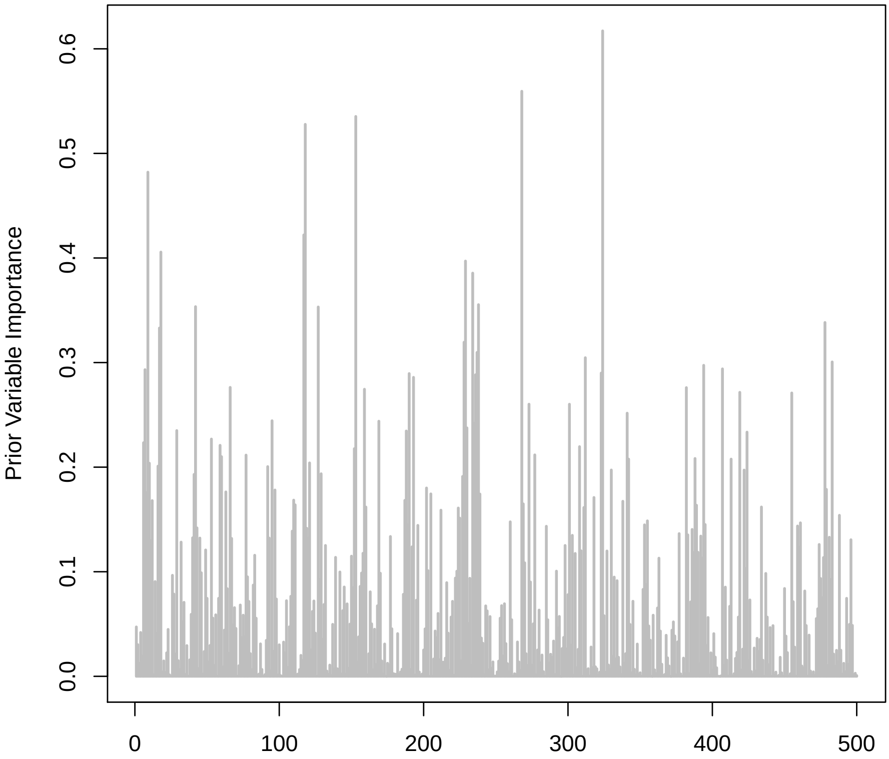

Figure 1. Prior Feature Importance: A representative simulation results for regression analysis on synthetic dataset of scenario with number of instances n=25, number of features p=500, correlation coefficient ρ=0.5, number of learners=450, and number of replications=100..

If the design orthonormal, then the base learner prediction reduces to

$$\widehat{\boldsymbol{\beta}}^{(l)} = (\boldsymbol{X}^{(l)})^{\top}\boldsymbol{Y}^{(l)}.$$

Here $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{Y} \in \mathbb{R}^{n\times 1}$, $(\boldsymbol{X}^{(l)})^{\top}\boldsymbol{Y}^{(l)} \in \mathbb{R}^{p\times 1}$ is an $p \times 1$ vector, and we must have

$$((\boldsymbol{X}^{(l)})^{\top}\boldsymbol{Y}^{(l)})^{\top} = ((\boldsymbol{X}_1^{(l)})^{\top}\boldsymbol{Y}^{(l)}, (\boldsymbol{X}_2^{(l)})^{\top}\boldsymbol{Y}^{(l)}, \cdots, (\boldsymbol{X}_p^{(l)})^{\top}\boldsymbol{Y}^{(l)})$$

where $\boldsymbol{X}_j^{(l)}$ is the jth column of the design matrix $\boldsymbol{X}^{(l)}$, the lth bootstrap replicate of $\boldsymbol{X}$. Therefore, under the orthonormal design, we must have

$$\begin{aligned} \widehat{g}^{(l)}(\mathbf{x}) &= \mathbf{x}^{\top}\widehat{\boldsymbol{\beta}}^{(l)} = \mathbf{x}^{\top}(\boldsymbol{X}^{(l)})^{\top}\boldsymbol{Y}^{(l)} \\ &= \mathbf{x}^{\top}((\boldsymbol{X}_1^{(l)})^{\top}\boldsymbol{Y}^{(l)}, (\boldsymbol{X}_2^{(l)})^{\top}\boldsymbol{Y}^{(l)}, \cdots, (\boldsymbol{X}_p^{(l)})^{\top}\boldsymbol{Y}^{(l)}) \\ &= \sum_{j=1}^{p} \mathrm{x}_j(\boldsymbol{X}_j^{(l)})^{\top}\boldsymbol{Y}^{(l)}. \end{aligned}$$

Now, $\boldsymbol{\gamma}^{(l)} = (\gamma_1^{(l)}, \gamma_2^{(l)}, \cdots, \gamma_p^{(l)})$ as in (3.1) is the lth bootstrap indicator vector. Since the prediction with a base learner only involves those variables that are

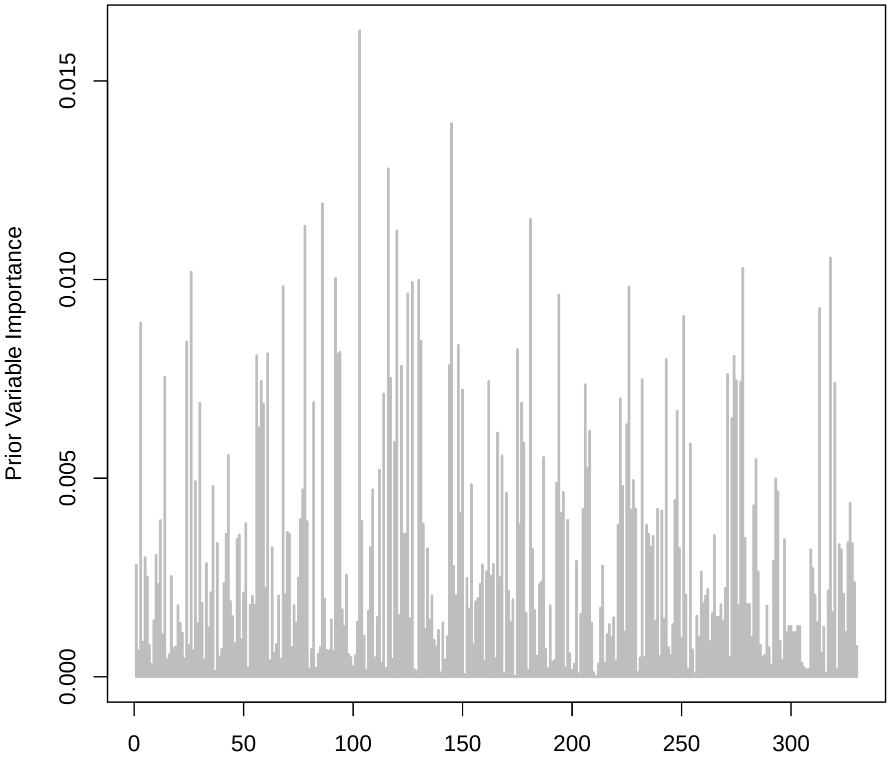

Figure 2. Prior Feature Importance: A representative simulation results for classification analysis on real dataset of Lymphoma disease..

active in the current bootstrapped subspace, we have

$$\mathrm{x}_j(\boldsymbol{X}_j^{(l)})^\top \boldsymbol{Y}^{(l)} = \begin{cases} \mathrm{x}_j(\boldsymbol{X}_j^{(l)})^\top \boldsymbol{Y}^{(l)} & \text{if } \gamma_j^{(l)} = 1 \\ 0 & \text{if } \gamma_j^{(l)} = 0 \ . \end{cases}$$

Now, the prediction of the response for $\mathbf{x}^*$ is therefore given by

$$\begin{aligned} \widehat{f}^{(L)}_{\texttt{RASSEL}}(\mathbf{x}^*) &= \frac{1}{L}\sum_{l=1}^{L}\widehat{g}^{(l)}(\mathbf{x}^*) = \frac{1}{L}\sum_{l=1}^{L}\sum_{j=1}^{p}\mathrm{x}_j^*(\boldsymbol{X}_j^{(l)})^\top \boldsymbol{Y}^{(l)} \\ &= \sum_{j=1}^{p}\mathrm{x}_j^*\left\{\frac{1}{L}\sum_{l=1}^{L}\gamma_j^{(l)}(\boldsymbol{X}_j^{(l)})^\top \boldsymbol{Y}^{(l)}\right\} \\ &= \sum_{j=1}^{p}\mathrm{x}_j^*\hat{\beta}_j^{(L)} \end{aligned}$$

where

$$\hat{\beta}_j^{(L)} = \frac{1}{L}\sum_{l=1}^{L}\gamma_j^{(l)}(\boldsymbol{X}_j^{(l)})^\top \boldsymbol{Y}^{(l)}.$$

□

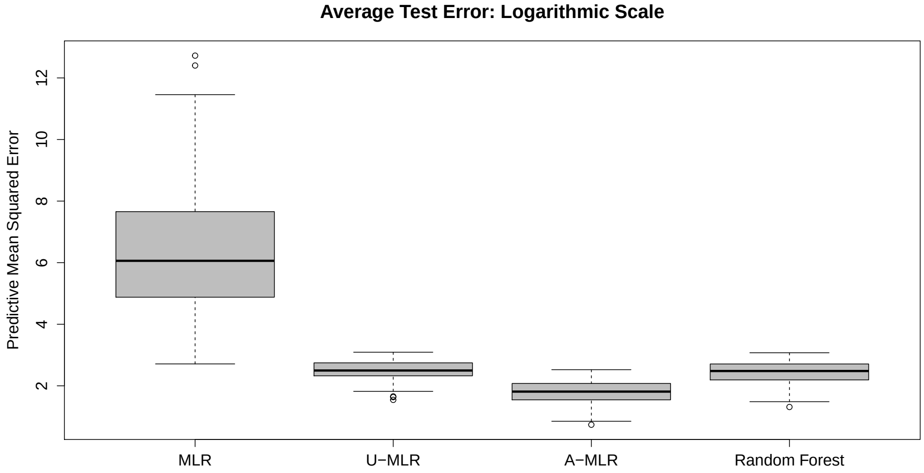

Figure 3. A representative results of synthetic dataset of scenario with number of instances n=50, number of features p=1000, correlation coefficient ρ=0.05, number of learners=450, and number of replications=100. We used the correlation weighting scheme for regression analysis on logarithmic scale. The abbreviations used in this figure are as follows: multiple linear regression (MLR), uniform multiple linear regression (U-MLR), adaptive-multiple linear regression (A-MLR).

7. Computational demonstrations

We used a collection of simulated and real-world datasets for our experiments. We report the mean square error (MSE) for each individual algorithm and task purposes, i.e., regression, or classification. We designed our artificial datasets to fit six scenarios based on the factors, which are the dimensionality of the data (number of features), the number of sample size (number of instances), and the correlation of the data. For the purposes of consistency and completeness, we choose the real datasets that carries different characteristics in terms of the number of instances and the number of features along with variety of applications.

7.1. Simulated example

The dataset in this example is simulated data with different scenarios on the level of correlation among the variables, and the ratio n and p. In this particular example, the true function is

$$f(\mathbf{x}) = 1 + 2\mathrm{x}_3 - 2\mathrm{x}_7 + 3\mathrm{x}_9$$

with $\mathbf{x} \sim \mathtt{MVN}(\mathbf{1}_9, \Sigma_\rho)$ and $\epsilon \sim \mathbf{N}(0, 2^2)$. Specifically, we simulate data by defining $\rho \in [0, 1)$, then we generate our predictor variables using a multivariate normal distribution. Throughout this paper, the multivariate Gaussian density will be denoted by $\phi_p(\mathbf{x}; \boldsymbol{\mu}, \Sigma)$

$$\phi_p(\mathbf{x}; \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^p|\Sigma|}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\}.$$

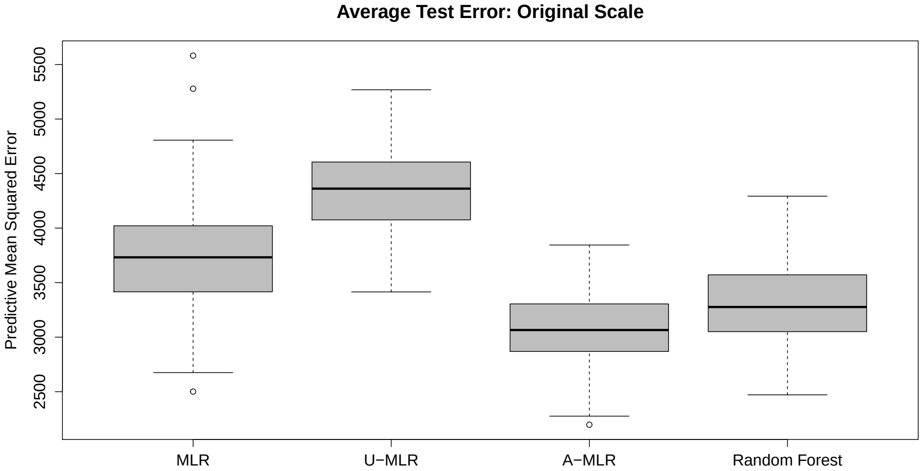

Figure 4. A representative results of Diabetes interaction real dataset with correlation weighting scheme for regression analysis on original scale. The abbreviations used in this figure are as follows: multiple linear regression (MLR), uniform multiple linear regression (U-MLR), adaptive-multiple linear regression (A-MLR).

Furthermore, in order to study the effect of the correlation pattern, we simulate the data using a covariance matrix Σ parameterized by τ and ρ and defined by $\tau\Sigma$ where $\Sigma = (\sigma_{ij})$ with $\sigma_{ij} = \rho^{|i-j|}$.

$$\Sigma = \Sigma(\tau, \rho) = \tau \begin{pmatrix} 1 & \rho & \cdots & \rho^{p-2} & \rho^{p-1} \\ \rho & 1 & \rho & \cdots & \rho^{p-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho^{p-2} & \ddots & \rho & 1 & \rho \\ \rho^{p-1} & \rho^{p-2} & \cdots & \rho & 1 \end{pmatrix}.$$

For simplicity however, we use the first Σ with $\tau = 1$ throughout this paper. For the remaining parameters, we use $\rho \in \{0.05, 0.5\}$ and $p \in \{25, 50, 250, 1000\}$, with $n \in \{25, 50, 250, 1000\}$. In addition, to check the robustness of our developed framework through measuring the average test error, we systematically change the correlation coefficient over its whole range, which is [0,1), with large p and small n. Therefore, we designed our artificial datasets to fit six scenarios based on the following factors: (a) the dimensionality of the data (number of features), (b) the sample size (number of instances), and (c) the correlation of the data. See Tables 1–4.

7.2. Real-life datasets

We choose real life DNA Microarray Gene Expression datasets (See Table 5) because it carry different characteristics in terms of the number of instances and features and IFR ratio. In addition, these real datasets carry hidden correlation between features, which makes the prediction problem very difficult.

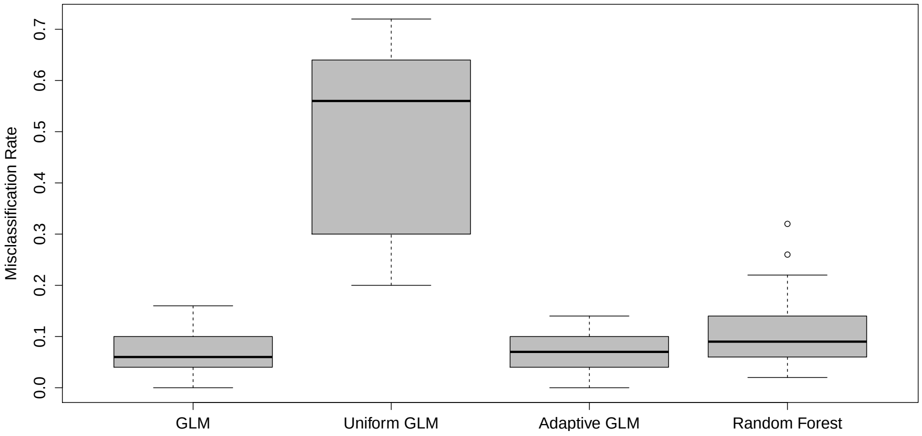

Figure 5. A representative results on synthetic dataset of scenario with number of instances n=200, number of features p=25, correlation coefficient ρ=0.05, number of learners=450, and number of replications=100. We used F-statistics weighting scheme for classification analysis..

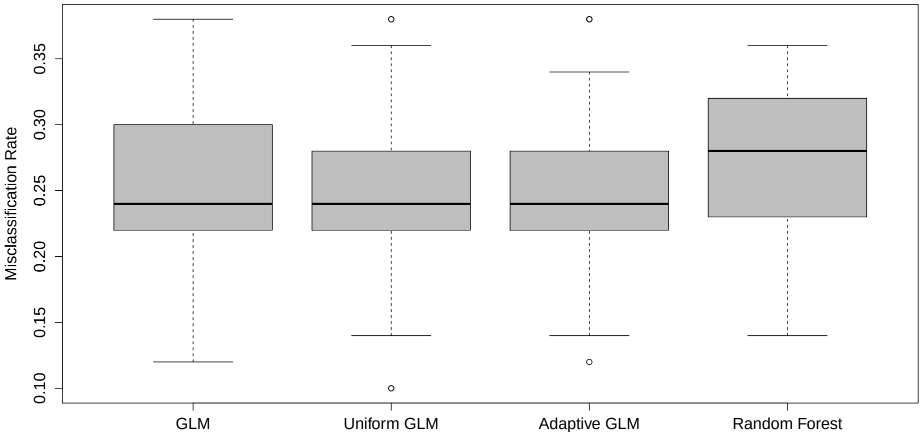

Figure 6. A representative results of the Diabetes in Pima Indian Women real dataset with F-statistics weighting scheme for classification analysis..

7.3. Quantitative analysis

Figs. 1 and 2 show the prior feature importance for both regression and classification purposes on synthetic and real datasets. As you can see from these figures that extracting important features is an important step for RASSEL algorithm. For regression analysis, the assessment of the performance of our developed framework was done through measuring the average mean square error (MSE) and as shown in Figs. 3 and 4 that the AMLR posses the smallest predictive MSE. For classification analysis, the evaluation of the performance of our developed framework through quantifying the average misclassification rate (MCR) and as shown

Table 1. Regression Analysis: Mean Square Error (MSE) for different machine learning algorithms on various scenarios of synthetic datasets..

Weighting	n	p	ρ	MLR	Uniform MLR	Adaptive MLR	RF	Better?
Correlation	200	25	0.05	5.69±0.89	14.50±2.63	4.60±0.706	9.81±1.86	AMLR
	200	25	0.5	4.78±0.81	11.67±2.55	4.77±0.94	8.46±1.97	AMLR
	25	200	0.05	974.37±5.e3	18.35±6.92	8.10±3.86	18.56±7.24	AMLR
	25	200	0.5	5.e3±5.e4	18.83±8.72	8.27±5.24	18.18±8.65	AMLR
	50	1000	0.05	2.e4±1.e5	28.36±11.51	12.38±5.91	27.92±11.78	AMLR
	1000	50	0.05	4.66±0.34	16.62±1.37	4.33±0.33	6.73±0.62	AMLR
F-statistics	200	25	0.05	5.04±0.79	14.42±2.67	4.48±0.74	8.75±1.76	AMLR
	200	25	0.5	4.49±0.76	12.06±2.04	5.51±1.09	8.33±1.59	MLR
	25	200	0.05	3.e4±2.e5	17.77±9.15	5.81±4.10	15.81±8.55	AMLR
	25	200	0.5	1.e4±1.e5	23.09±16.06	12.53±10.27	24.11±16.31	AMLR
	50	1000	0.05	4.e5±3.e6	16.65±5.38	7.65±2.83	15.54±5.31	AMLR
	1000	50	0.05	4.19±0.33	15.97±1.15	3.90±0.30	6.24±0.55	AMLR

Table 2. Regression Analysis: Mean Square Error (MSE) for different machine learning algorithms on real datasets..

Data Set	Weighting	MLR	Uniform MLR	Adaptive MLR	RF	Better?
BodyFat	correlation	17.41±2.69	23.59±3.71	19.25±3.06	19.72±3.18	MLR
	F-statistics	17.06±2.50	23.07±3.46	17.46±2.65	19.51±2.99	MLR
Attitude	correlation	74.12±32.06	80.35±34.40	58.49±20.21	88.72±35.97	AMLR
	F-statistics	75.19±36.63	74.71±33.17	51.84±15.19	82.21±35.58	AMLR
Cement	correlation	10.76±7.25	NA	19.92±15.98	75.91±56.05	MLR
	F-statistics	11.07±8.55	NA	24.27±18.27	62.20±46.53	MLR
Diabetes 1	correlation	2998.13±322.37	3522.30±311.81	3165.74±300.86	3203.94±311.94	MLR
	F-statistics	2988.32±341.20	3533.45±375.38	3133.60±324.75	3214.11±318.6931	MLR
Diabetes 2	correlation	3916.98±782.35	4244.00±390.29	3016.54±285.89	3266.50±324.82	AMLR
	F-statistics	3889.00±679.55	4306.76±419.66	3076.77±338.08	3326.28±382.37	AMLR
Longley	correlation	0.21±0.13	0.62±0.36	0.49±0.29	1.54±0.92	MLR
	F-statistics	0.22±0.13	0.66±0.42	0.49±0.29	1.63±1.04	MLR

Table 3. Classification Analysis: MisClassification Rate (MCR) for different machine learning algorithms on various scenarios of simulated datasets..

Weighting	n	p	ρ	GLM	Uniform GLM	Adaptive GLM	RF	Better?
F-statistics	200	25	0.05	0.070±0.033	0.486±0.172	0.071±0.032	0.101±0.053	AGLM
	200	25	0.5	0.140±0.045	0.498±0.221	0.138±0.043	0.136±0.058	RF
	50	200	0.05	0.102±0.093	0.673±0.123	0.100±0.092	0.320±0.103	AGLM
	50	200	0.5	0.058±0.141	0.346±0.346	0.049±0.121	0.178±0.188	AGLM
	50	1000	0.05	0.033±0.064	0.522±0.158	0.034±0.062	0.409±0.114	AGLM
	1000	50	0.05	0.130±0.019	0.643±0.028	0.130±0.019	0.167±0.024	AGLM

in Figs. 5 and 6 that the AGLM outperforms RF and has the same MCR with both GLM and UGLM.

8. Discussion

Based on simulation results (Tables 1–4), which performed on both synthetic and real datasets, that choosing which weighting scheme used is crucial and can affect the accuracy of the RASSEL algorithm. The ideal example has been shown in

Table 4. Classification Analysis: MisClassification Rate (MCR) for different machine learning algorithms on real datasets..

Data Set	W. S.	GLM	Uni. GLM	Adap. GLM	RF	Better?
Diabetes in Pima	F-stat	0.274±0.071	0.249±0.051	0.255±0.051	0.269±0.050	AGLM
Prostate Cancer	F-stat	0.425±0.113	0.355±0.093	0.332±0.094	0.343±0.098	AGLM
Golub Leukemia	F-stat	0.427±0.116	0.023±0.103	0.021±0.011	0.023±0.013	AGLM
Diabetes	F-stat	0.034±0.031	0.068±0.039	0.038±0.034	0.031±0.029	RF
Lymphoma	F-stat	0.248±0.065	0.057±0.034	0.046±0.029	0.082±0.046	AGLM
Lung Cancer	F-stat	0.113±0.051	0.038±0.023	0.037±0.024	0.051±0.030	AGLM
Colon Cancer	F-stat	0.296±0.124	0.168±0.095	0.124±0.074	0.199±0.106	AGLM

Table 5. Summary of the regression and classification real datasets..

Data set	# instan.	# features	IFR ratio
regression			
Bodyfat	252	14	1,800.00%
attitude	30	7	428.50%
Cement	13	5	260.00%
Diabetes 1	442	11	4,018.00%
Diabetes 2	442	65	680.00%
Longley	16	7	228.50%
classification			
Diabetes in Pima	200	8	2,500.00%
Prostate cancer	79	501	15.80%
Golub Leukemia	72	3572	2.00%
Diabetes	145	4	3,625.00%
Lymphoma	180	662	27.20%
Lung Cancer	197	1,000	19.70%
Colon Cancer	62	2,000	3.10%

Table 1 where the using F-statistics instead of correlation coefficient as a weighting scheme degrades the AMLR performance and makes the MLR achieves less MSE. Also, as revealed from our simulations on generated synthetic datasets that when the number of selected features is higher than 15–20, our proposed framework yields ensemble classifiers, which are highly stable and very accurate. The idea behind it is when the number of voters is large enough, the random process of attribute selection yields sufficient number of different classifiers, which ensure high accuracy and stability for ensemble learning procedure. Moreover, as the number of features increase, the performance of the RASSEL algorithm stays strong. As an example, the accuracy of our developed framework is $\sim$1.8 larger than the random forest for Lymphoma dataset, that has 662 features, and it outperform RF for the Leukemia dataset, which possess over 3,500 features. The same pattern was noticed by [1]. Regarding testing the RASSEL algorithm on correlated datasets, we perform simulations on synthetic datasets with changing the correlation coefficient (ρ) between [0,1). As depicted in Fig. 7 that AMLR

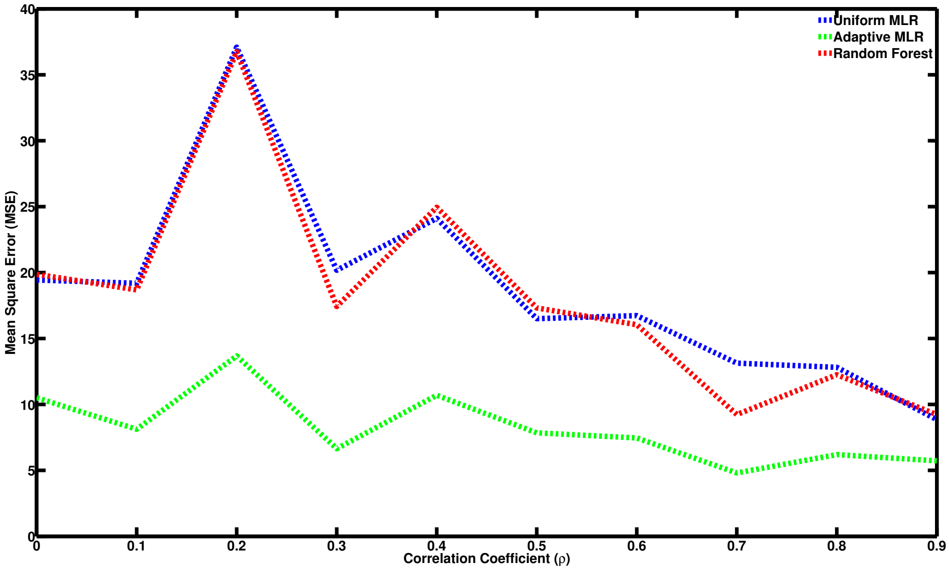

Figure 7. A representative results that exhibits the relationship between mean square error (MSE) and correlation coefficient (ρ) for different algorithms on synthetic dataset with correlation weighting scheme for regression analysis when p≫n..

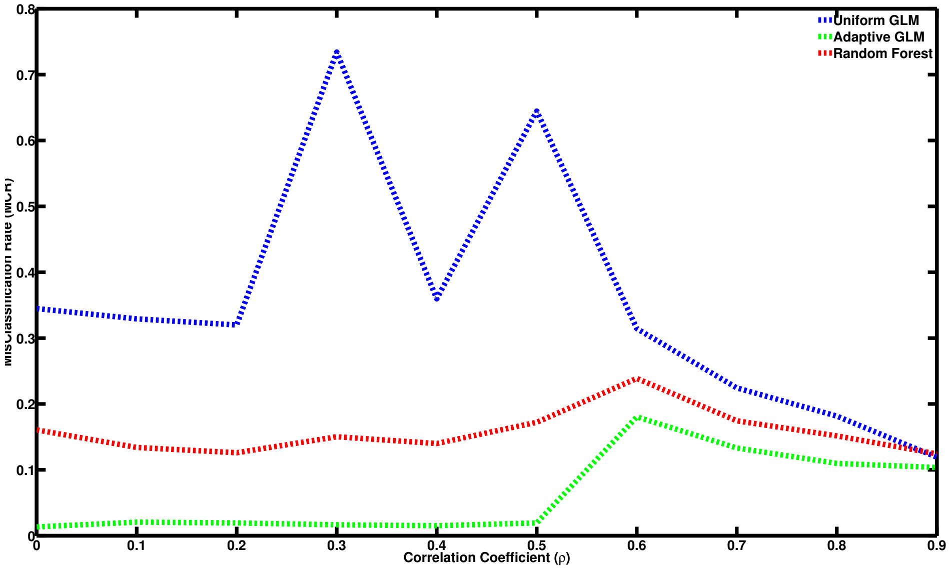

Figure 8. A representative results that exhibits the relationship between mean square error (MSE) and correlation coefficient (ρ) for different algorithms on synthetic dataset with F-statistics weighting scheme for classification analysis when n≫p..

posses the lowest MSE constantly and both UMLR and RF have the same MSE.

Also, Fig. 8 shows that the AGLM outperforms both UGLM and RF and has almost zero MSE when ρ <0.5.

In the case of correlated variables, we use the correlation matrix to dynamically modify the π_j's in a manner similar to stickbreaking use in nonparametric Dirichlet process construction. This extension of our work is addressed in a subsequent paper in preparation.

Even though our developed adaptive RASSEL algorithm outperforms many classifier ensembles on almost all the computations explored in this paper, it has limitations. For instance, our method can not deal with dataset that has categorical features. Instead it necessities to encode these features numerically. In addition, our algorithm fails to select the optimal feature subsets, when the number of features are very small.

9. Conclusion and future work

We performed a data-driven quantitative analysis of the developed adaptive RASSEL algorithm for an ensemble prediction problem. We present a rigorous theoretical justification of our propose algorithm as well as empirically through performing simulations on synthetic and real datasets. The key important issues for the developed algorithm resides on four fundamental factors: (a) Generalization: any base learner can be adapted easily. (b) Flexibility: a straightforward data-driven weighting scheme can be used in any supervised learning scheme. (c) Speed: reduced computational complexity, which is necessary in other ensemble learning algorithms, such as the permutation step need in RF. (d) Accuracy: RASSEL achieves less MSE compared with the most known ensemble learning algorithm, i.e. RF.

For now, we choose fixed number of attribute subset. However, the algorithm should evaluated based on the performance to determine the appropriate number for single classifiers used in the ensemble learning. Therefore, we plan on implementing an extension of our framework whereby the dimension of the subspace may be adaptively updated. Also, given the availability of computing power, we plan to use other techniques such as cross validation to determine the optimal number of base learners to include in the ensemble learning. Finally, the adaptive RASSEL algorithm is tested on a relatively small datasets. So, our next step will be applying the developed algorithm on a big datasets.

References

[1] D. Amaratunga, J. Cabrera and Y.-S. Lee, *Enriched random forests*, Bioinformatics **24** (2008), 2010–2014.

[2] A. Bertoni, R. Folgieri and G. Valentini, *Bio-molecular cancer prediction with random subspace ensembles of support vector machines*, Neurocomputing **63** (2005), 535–539.

[3] L. Breiman, *Bagging predictors*, Machine Learning **24** (1996), 123–140.

[4] L. Breiman, *Random forests*, Machine Learning **45** (2001), 5–32.

[5] M. Denil, D. Matheson and N. de Freitas, *Narrowing the gap: Random forests in theory and in practice*, in: E. P. Xing and T. Jebara (eds.), Proceedings of the 31st International Conference on Machine Learning, PMLR **32**, 2014, 665–673.

[6] R. Díaz-Uriarte and S. A. de Andrés, *Gene selection and classification of microarray data using random forest*, BMC Bioinformatics **7** (2006), 13 pp.

[7] Y. Freund, *Boosting a weak learning algorithm by majority*, Information and Computation, **121** (1995), 256–285.

[8] Y. Freundand and R. E. Schapire, *A Decision-theoretic generalization of on-line learning and an application to boosting*, Journal of Computer and System Sciences **55** (1997), 119–139.

[9] Y. Freund and R. E. Schapire, *Experiments with a new boosting algorithm*, in: L. Saitta (ed.), Proceeding ICML'96 Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, 1996, 148–156.

[10] P. Geurts, D. Ernst and L. Wehenkel, *Extremely randomized trees*, Machine Learning **63** (2006), 3–42.

[11] B. Goldstein, A. E. Hubbard, A. Cutler and L. F. Barcellos, *An application of random forests to a genome-wide association dataset: Methodological considerations and new findings*, BMC Genetics **11**:49 (2010), 13 pp.

[12] N. Gunduz and E. Fokoue, *Robust classification of high dimension low sample size data*, arXiv:1501.00592v1 [stat.AP], 2015, 17 pp.

[13] L K. Hansen and P. Salamon, *Neural network ensembles*, IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990), 993–1001.

[14] T. K. Ho, *The random subspace method for constructing decision forests*, IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998), 832–844.

[15] A. Joly, P. Geurts and L. Wehenkel, *Random forests with random projections of the output space for high dimensional multi-label classification*, in: T. Calders, F. Esposito, E. Hüllermeier and R. Meo (eds.), Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science **8724**, Springer, Berlin Heidelberg, 2014, 607–622.

[16] L. Kuncheva, J. Rodriguez, C. Plumpton, D. Linden and S. Johnston, *Random subspace ensembles for fMRI classification*, IEEE Transactions on Medical Imaging **29** (2010), 531–542.

[17] B.H. Menze, B. Kelm, D. Splitthoff, U. Koethe and F. Hamprecht, *On oblique random forests*, in: D. Gunopulos, T. Hofmann, D. Malerba and M. Vazirgiannis (eds.), Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science **6912**, Springer, Berlin Heidelberg, 2011, 453–469.

[18] P. Panov and S. Džeroski, *Combining bagging and random subspaces to create better ensembles*, in: M. R. Berthold, J. Shawe-Taylor and N. Lavrač (eds.), Advances in Intelligent Data Analysis VII, Lecture Notes in Computer Science **4723**, Springer, Berlin Heidelberg, 2007, 118–129.

[19] K. Tumer and J. Ghosh, *Classifier combining: Analytical results and implications*, in: Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms, AAAI Press, 1995, 126–132.

[20] K. Tumer and N. C. Oza, *Decimated input ensembles for improved generalization*, IJCNN '99, Neural Networks, 1999, 3069–3074.

[21] M. Van Wezel and R. Potharst, *Improved customer choice predictions using ensemble methods*, European Journal of Operational Research **181** (2007), 436–452.

[22] D. H. Wolpert, *Stacked generalization*, Neural Networks **5** (1992), 241–259.

[23] Y. Ye, Q. Wu, J. Z. Huang, M. K. Ng and X. Li, *Stratified sampling for feature subspace selection in random forests for high dimensional data*, Pattern Recognition **46** (2013), 769–787.

Mohamed Elshrif, Qatar Computing Research Institute (QCRI), Hamad Bin Khalifa University (HBKU), Doha, Qatar
e-mail: `melshrif@hbku.edu.qa`

Ernest Fokoué, School of Mathematical Sciences, Rochester Institute of Technology (RIT), Rochester, NY 14623 USA
e-mail: `epfeqa@rit.edu`