Electronic Journal of Statistics
Vol. 3 (2009) 1288–1304
ISSN: 1935-7524
DOI: 10.1214/09-EJS434

Forest Garrote

Nicolai Meinshausen

Department of Statistics, University of Oxford
1 South Parks Road, OX1 3TG, UK
e-mail: meinshausen@stats.ox.ac.uk

Abstract: Variable selection for high-dimensional linear models has received a lot of attention lately, mostly in the context of ℓ_1-regularization. Part of the attraction is the variable selection effect: parsimonious models are obtained, which are very suitable for interpretation. In terms of predictive power, however, these regularized linear models are often slightly inferior to machine learning procedures like tree ensembles. Tree ensembles, on the other hand, lack usually a formal way of variable selection and are difficult to visualize. A Garrote-style convex penalty for trees ensembles, in particular Random Forests, is proposed. The penalty selects functional groups of nodes in the trees. These could be as simple as monotone functions of individual predictor variables. This yields a parsimonious function fit, which lends itself easily to visualization and interpretation. The predictive power is maintained at least at the same level as the original tree ensemble. A key feature of the method is that, once a tree ensemble is fitted, no further tuning parameter needs to be selected. The empirical performance is demonstrated on a wide array of datasets.

AMS 2000 subject classifications: 62G08.
Keywords and phrases: Nonnegative Garrote, Random Forests, sparsity, tree ensembles.

Received June 2009.

1. Introduction

Given data (X_i, Y_i), for $i = 1, \ldots, n$, with a p-dimensional real-valued predictor variable X, where $X = (X^{(1)}, \ldots, X^{(p)}) \in \mathcal{X}$, and a real-valued response Y, a typical goal of regression analysis is to find an estimator $\hat{Y}(x)$, such that the expected loss $E\big(L(\hat{Y}(X), X)\big)$ is minimal, under a given loss function $L : \mathcal{X} \times \mathbb{R} \mapsto \mathbb{R}^+$. For the following, the standard squared error loss is used. If the predictor can be of the 'black-box' type, tree ensembles have proven to be very powerful. Random Forests (Breiman, 2001) is a prime example. Random Forests are essentially bagged regression or classification trees, with the added twist that the variable to split on is searched for each node in each tree only over a randomly selected subset of all variables. Another example of tree ensembles are boosted regression trees (Yu and Bühlmann, 2003). There are many interesting tools available for interpretation of these tree ensembles, see for example Strobl *et al.* (2007, 2008) and the references therein.

While tree ensembles often have very good predictive performance, an advantage of a linear model is better interpretability. Measuring variable importance and performing variable selection is easier to formulate and understand in the

context of linear models. For high-dimensional data with $p \gg n$, regularization is clearly imperative and the Lasso (Tibshirani, 1996; Chen, Donoho and Saunders, 2001) has proven to be very popular in recent years, since it combines a convex optimization problem with variable selection. Lasso is estimating a linear model for a given penalty parameter $\lambda \geq 0$ by

$$\hat{\beta}^{\lambda} \;=\; \text{argmin}_{\beta \in \mathbb{R}^p} \; \sum_{i=1}^{n} \left(Y_i - \sum_{k=1}^{p} \beta_k X_i^{(k)} \right)^2 \;+\; \lambda \|\beta\|_1,$$

where $\|\beta\|_1 = \sum_{k=1}^{p} |\beta_k|$ is the ℓ_1-norm of the coefficient vector $\beta = (\beta_1, \ldots, \beta_k)$. The penalty parameter is chosen typically by cross-validation. A precursor to the Lasso was the nonnegative Garrote (Breiman, 1995). Based on an initial estimator $\tilde{\beta}$ (for example the least squares estimator if available), the nonnegative Garrote estimator $\hat{\beta}^c$ is defined under a constraint $0 < c \leq p$ as $\hat{\beta}^c = \hat{\gamma}\tilde{\beta}$, where $(\hat{\gamma}\tilde{\beta})_k = \hat{\gamma}_k \tilde{\beta}_k$ for all $k = 1, \ldots, p$. The shrinkage vector $\hat{\gamma} = (\hat{\gamma}_1, \ldots, \hat{\gamma}_p)$ is found as

$$\hat{\gamma} = \text{argmin}_{\gamma \in \mathbb{R}^p} \; \sum_{i=1}^{n} \left(Y_i - \sum_{k=1}^{p} \gamma_k \tilde{\beta}_k X_i^{(k)} \right)^2 \text{ such that } \min_k \gamma_k \geq 0 \text{ and } \sum_{k=1}^{p} \gamma_k \leq c.$$

Alternatively, the estimator can be written in penalized form, adding the ℓ_1-constraint on γ as a penalty factor to the objective function, in analogy to the form of the Lasso estimator given above. A disadvantage of the nonnegative Garrote is the reliance on an initial estimator, which could be the least squares estimator (if $n < p$) or a regularized estimator. On the positive side, important variables incur less penalty and bias under the regularization than they do with the Lasso. For a deeper discussion of the properties of the nonnegative Garrote see Yuan and Lin (2007).

Here, it is proposed to use Random Forest as an initial estimator for the nonnegative Garrote. The idea is related to the Rule Ensemble approach of Friedman and Popescu (2008), who used the Lasso instead of the nonnegative Garrote. A crucial distinction is that rules fulfilling the same functional role are grouped in our approach. This is similar in spirit to the group Lasso (Meier, van de Geer and Buhlmann, 2008; Yuan and Lin, 2006; Zhao, Rocha and Yu, 2009). This produces a very accurate predictor that uses just a few functional groups of rules, discarding many variables in the process as irrelevant.

A unique feature of the proposed method is that is seems to work very well in the absence of a tuning parameter. It just requires the choice of an initial tree ensemble. This makes the procedure very simple to implement and computationally efficient. The idea and the algorithm is developed in Section 2, while a detailed numerical study on 15 datasets makes up Section 3.

2. Methods

2.1. Trees and equivalent rules

A tree T is seen here as a piecewise-constant function $\mathbb{R}^p \mapsto \mathbb{R}$ derived from the tree structure in the sense of Breiman *et al.* (1984). Friedman and Popescu (2008) proposed 'rules' as a name for simple rectangular-shaped indicator functions. Every node j in a tree is associated with a B_j in $\mathbb{R}^p$-dimensional space, defined as the set of all values $x \in \mathbb{R}^p$ that pass through node j if passed down the tree. All values $x \in \mathbb{R}^p$ that do not pass through node j are outside of B_j. The way rules are used here, they correspond to indicator functions

$$R_j(x) = \begin{cases} 1 & x \in B_j \\ 0 & x \notin B_j \end{cases},$$

i.e. $R_j(x) = 1\{x \in B_j\}$ is the indicator function for box B_j. For a more detailed discussion see Friedman and Popescu (2008).

To give an example of a rule, take the well-know dataset on abalone (Nash *et al.*, 1994). The goal is to predict age of abalone from physical measurements. For each of the 4177 abalone in the dataset, eight predictor variables (sex, length, diameter, height, while weight, shucked weight, viscera weight and shell weight) are available. An example of a rule is

$$R_1(x) = 1\{\text{diameter} \geq 0.537 \text{ and shell weight} \geq 0.135\}, \tag{1}$$

and the presence of such a rule in a final predictor is easy to interpret, comparable to interpreting coefficients in a linear model.

Every regression tree can be written as a linear superposition of rules. Suppose a tree T has J nodes in total. The regression function $\hat{T}$ of this tree (ensemble) can then be written as

$$\hat{T}(x) = \sum_{j=1}^{J} \hat{\beta}_j^{tree} R_j(x) \tag{2}$$

for some $\hat{\beta}^{tree}$. The decomposition is not unique in general. We could, for example, assign non-zero regression coefficients $\hat{\beta}_j$ only to leaf nodes. Here, we build the regression function incrementally instead, assigning non-zero regression coefficients to *all* nodes. The coefficient $\hat{\beta}_j^{tree}$ is supposed to measure, in a sense, the strength of the signal in the data picked up by rule R_j that was not already picked up by the parent node of R_j. The coefficients $\hat{\beta}_j^{tree}$ are defined as

$$\hat{\beta}_j^{tree} = \begin{cases} E_n(Y|x \in B_j) - E_n(Y|x \in B_{pa(j)}) & \text{if } j \text{ is not root node} \\ E_n(Y) & \text{if } j \text{ is root node} \end{cases}, \tag{3}$$

where E_n is the empirical mean across the n observations and $pa(j)$ is the parent node of j in the tree. Definition (3) is valid for single trees. For tree ensembles, the regression coefficient (3) is first computed separately for all nodes/rules in each tree. A given rule R_j does in general not appear in all trees and coefficient

(3) is defined to be 0 for all trees in which R_j does not appear. The overall regression coefficient for each rule $_j$ is then the average of (3) across all trees in the ensemble.

Rule (1), in the abalone example above, receives a regression weight $\hat{\beta}_1^{tree} = 0.0237$. The contribution of rule (1) to the Random Forest fit is thus to increase the fitted value if and only if the diameter is larger than 0.537 and shell weight is larger than 0.135. The Random Forest fit is the sum of the contribution from all these rules, where each rule corresponds to one node in the tree ensemble.

To see that (2) and (3) really correspond to the original tree (ensemble) solution, consider just a single tree for the moment. Denote the predicted value for predictor variable X by $\hat{T}(x)$. Let $B_{leaf(x)}$ be the rectangular area in p-dimensional space that corresponds to the leaf node $leaf(x)$ of the tree in which predictor variable X falls. The predicted value follows then by adding up all relevant nodes and obtaining with (2) and (3),

$$\hat{T}(x) = E_n(Y|x \in B_{leaf(x)}),$$

since each $x \in \mathcal{X}$ falls into several nodes as it passes through a tree, including the root node and one leaf node. The contribution of the second term $-E_n(Y|x \in B_{pa(j)})$ of each node j apart from the root node in (3) cancels exactly the contribution of the first term $E_n(Y|x \in B_j)$ for the corresponding parent node. The only term surving is hence the contibution $E_n(Y|x \in B_{leaf})$ of the leaf node in which observation x falls. The predicted value is thus just the empirical mean of Y across all observations $i = 1, \ldots, n$ which fall into the same leaf node as X, which is equivalent to the original prediction of the tree. The equivalence for tree ensembles follows by averaging across all individual trees as in (2).

A similar argument could be made to work for binary prediction with class labels $Y \in \{0, 1\}$, but decomposition (3) is more natural for regression and this will be the sole focus below.

For the following, it is assumed that rules contain at most a single inequality for each variable. In other words, the boxes defined by rules in p-dimensional spaces are defined by at most a single hyperplane in each variable. If a rule violates this assumption, it contains one (or more) variables with two hyperplanes in the rule definition and can be decomposed into two or several rules satisfying the assumption. For any such rule R_j, there will be some $k \in \{1, \ldots, p\}$ such that the rule is of the form

$$1\{c_{\min} \le x^{(k)} \le c_{\max}, A\},$$

where A is an event that involves splits on variables other than variable k. The contribution of this rule to the total fit, according to (3), can be decomposed in either of two ways as

$$\begin{aligned} \hat{\beta}_j 1\{c_{\min} \le x^{(k)} \le c_{\max}, A\} &= \hat{\beta}_j 1\{x^{(k)} \ge c_{\min}, A\} - \hat{\beta}_j 1\{x^{(k)} > c_{\max}, A\} \\ &= -\hat{\beta}_j 1\{x^{(k)} < c_{\min}, A\} + \hat{\beta}_j 1\{x^{(k)} \le c_{\max}, A\} \end{aligned} \tag{4}$$

and we replace the contribution of rule R_j in the tree decomposition (2), the left hand side in (4), by either of the two decompositions on the right hand side of (4). The final estimator is unaffected by the choice since the two functions $\hat{\beta}_j 1\{x^{(k)} \geq c_{\min}, A\}$ and $-\hat{\beta}_j 1\{x^{(k)} < c_{\min}, A\}$ are actually identical except for a shift in mean and the same holds true for the other two indicator functions. If the event or condition A contains further variables with two splitpoints, the process can be iterated until all rules in (2) involve only variables with a single split.

2.2. Rule ensembles

The idea of Friedman and Popescu (2008) is to modify the coefficients $\hat{\beta}^{tree}$ in (2), increasing sparsity of the fit by setting many regression coefficient to 0 and eliminating the corresponding rules from the fit, while hopefully not degrading the predictive performance of the tree ensemble in the process. The rule ensemble predictors are thus of the form

$$\sum_{j=1}^{J} \hat{\beta}_j R_j(x). \tag{5}$$

Sparsity is enforced by penalizing the ℓ_1-norm of $\hat{\beta}$ in LASSO-style (Tibshirani, 1996),

$$\hat{\beta}^{\lambda} = \text{argmin}_{\beta} \sum_{i=1}^{n} \left(Y_i - \sum_{j=1}^{J} \beta_j R_j(x_i) \right)^2 \quad \text{such that} \quad \sum_{j=1}^{J} |\beta_j| \leq \lambda. \tag{6}$$

This enforces sparsity in terms of rules, i.e. the final predictor will have typically only very few rules, at least compared to the original tree ensemble. The penalty parameter λ is typically chosen by cross-validation. Friedman and Popescu (2008) recommend to add the linear main effects of all variables into (6), which were omitted here for notational simplicity and to keep invariance with respect to monotone transformations of predictor variables. It is shown in Friedman and Popescu (2008) that the rule ensemble estimator maintains in general the predictive ability of Random Forests, while lending itself more easily to interpretation.

2.3. Functional grouping of rules

The rule ensemble approach is treating all rules equally by enforcing a ℓ_1-penalty on all rules extracted from a tree ensemble. It does not take into account, however, that there are typically many very closely related rules in the fit. Take the RF fit to the abalone data as an example. Several hundred rules are extracted from the RF, two of which are

$$R_1(x) = 1\{\text{diameter} \geq 0.537 \text{ and shell weight} \geq 0.135\}, \tag{7}$$

with regression coefficient $\hat{\beta}_1 = 0.023$ and

$$R_2(x) = 1\{\text{diameter} \geq 0.537 \text{ and shell weight} \geq 0.177\}, \tag{8}$$

with regression coefficient $\hat{\beta}_2 = 0.019$. The effect of these two rules, measured by $\hat{\beta}_1 R_1$ and $\hat{\beta}_2 R_2$, are clearly very similar. In total, there are 32 rules 'interaction' rules that involve variables diameter and shell weight in the RF fit to the abalone data. Selecting some members of this group, it seems artificial to exclude others of the same 'functional' type.

Sparsity is measured purely on a rule-by-rule basis in the ℓ_1-penalty term of rule ensembles (6). Selecting the two rules mentioned above incurs the same sparsity penalty as if the second rule involved two completely different variables. An undesirable side-effect of not taking into account the grouping of rules is that many or even all original predictor variables might still be involved in the rules; sparsity is not explicitly enforced in the sense that many irrelevant original predictor variables are completely discarded in the selected rules.

It seems natural to let rules form functional groups. The question then turns up which rules form useful and interpretable groups. There is clearly no simple right or wrong answer to this question. Here, a very simple yet hopefully intuitive functional grouping of rules is employed. For the j-th rule, with coefficient $\hat{\beta}_j$, define the interaction pattern $\sigma_j = (\sigma_{j,1}, \ldots, \sigma_{j,p})$ for variables $k = 1, \ldots, p$ by

$$\sigma_{j,k} = \begin{cases} +1 & \text{iff } \sup_{x,x' \in \mathbb{R}^p : x^{(k)} > x'^{(k)}} \hat{\beta}_j \left(R_j(x) - R_j(x')\right) > 0 \\ -1 & \text{iff } \inf_{x,x' \in \mathbb{R}^p : x^{(k)} > x'^{(k)}} \hat{\beta}_j \left(R_j(x) - R_j(x')\right) < 0 \\ 0 & \text{otherwise} \end{cases} . \tag{9}$$

The meaning of interaction patterns is best understood if looking at examples for varying degrees, where the degree of a interaction pattern σ is understood to be the number of non-zero entries in σ and corresponds to the number of variables that are involved in a rule.

First degree (main effects). The simplest interaction patterns are those involving a single variable only, which correspond in some sense to the main effects of variables. The interaction pattern

$$(0, +, 0, 0, 0, 0, 0, 0)$$

for example collects all rules that involve the 2nd predictor variable *length* only and lead to a monotonically increasing fit (are thus of the form $1\{\text{length} \leq u\}$ for some real-valued u if the corresponding regression coefficient were positive or $1\{\text{length} \geq u\}$ if the regression coefficient were negative). The interaction pattern $(0, 0, -, 0, 0, 0, 0, 0)$ collects conversely all those rules that yield a monotonically decreasing fit in the variable *diameter*, the third variable.

Second degree (interactions effects). Second degree interaction patterns are of the form (7) or (8). As diameter is the 3rd variable and shell weight the 8th, the interaction pattern of both rules (7) and (8) is

$$(0, 0, +, 0, 0, 0, 0, +),$$

making them members of the same functional group, as for both rules the fitted value is monotonically increasing in both involved variables. In other words, either a large value in both variables increases the fitted value or a very low value in both variables decreases the fitted value. Second degree interaction patterns thus form four categories for each pair of variables. A case could be made to merge these four categories into just two categories, as the interaction patterns do not conform nicely with the more standard multiplicative form of interactions in linear models. However, there is no reason to believe that nature always adheres to the multiplicative form of interactions typically assumed in linear models. The interaction patterns used here seemed more adapted to the context of rule-based inference. Factorial variables can be dealt with in the same framework (9) by converting to dummy variables first.

2.4. Garrote correction and selection

In contrast to the group Lasso approach of Yuan and Lin (2006), the proposed method does not only start with knowledge of natural groups of variables or rules. A very good initial estimator is available, namely the Random Forest fit. This is exploited in the following.

Let $\hat{T}_\sigma$ be the part of the fit that collects all contributions from rules with interaction pattern σ,

$$\hat{T}_\sigma(x) = \sum_{j:\sigma(\hat{\beta}_j R_j)=\sigma} \hat{\beta}_j R_j(x). \tag{10}$$

Let $\mathcal{G}$ be the collection of all possible interaction patterns σ. The tree ensemble fit (2) can then be re-written as a sum over all interaction patterns

$$\hat{T}(x) = \sum_{\sigma\in\mathcal{G}} \hat{T}_\sigma(x). \tag{11}$$

A interaction pattern σ is called *active* if the corresponding fit in the tree ensemble is non-zero, i.e. if and only if $\hat{T}_\sigma$ is not identically 0. The Random Forest fit contains very often a huge number of active interaction pattern, involving interactions up to fourth and higher degrees. Most of those active patterns contribute just in a negligible way to the overall fit.

The idea proposed here is to use (11) as a starting point and modify it to enforce sparsity in the final fit, getting rid of as many unnecessary predictor variables and associated interaction patterns as possible. The Lasso of Tibshirani (1996) was used in the rule ensemble approach of Friedman and Popescu (2008). Here, however, the starting point is the functional decomposition (11), which is already a very good initial (yet not sparse) estimator of the underlying regression function.

Hence it seems more appropriate to use Breiman's nonnegative Garrote (Breiman, 1995), penalizing contributions of interaction patterns less if their contribution to the initial estimator is large and vice versa. The beneficial effect

of this bias reduction for important variables has been noted in, amongst others, Yuan and Lin (2007) and Zou (2006).

The Garrote-style Forest estimator $\hat{T}^{gar}$ is defined as

$$\hat{T}^{gar} = \sum_{\sigma \in \mathcal{G}} \gamma_\sigma \hat{T}_\sigma, \tag{12}$$

where $\gamma = (\gamma_{\sigma_1}, \ldots, \gamma_{\sigma_{|\mathcal{G}|}})$ is the vector of multiplicative weights. Each contribution $\hat{T}_\sigma$ of an interaction pattern σ is multiplied by the corresponding factor $\hat{\gamma}_\sigma$. The original tree ensemble fit in (11) is obtained by setting all factors equal to 1.

The multiplicative factor $\gamma = (\gamma_{\sigma_1}, \ldots, \gamma_{\sigma_{|\mathcal{G}|}})$ is chosen by least squares, subject to the constraint that the total ℓ_1-norm of the multiplying coefficients is less than 1,

$$\begin{aligned} \hat{\gamma} &= \operatorname{argmin}_{\gamma \in \mathbb{R}^{|\mathcal{G}|}} \sum_{i=1}^{n} \left(Y_i - \sum_{\sigma \in \mathcal{G}} \gamma_\sigma \hat{T}_\sigma(X_i) \right)^2 \\ &\quad \text{such that } |\mathcal{G}|^{-1} \sum_{\sigma \in \mathcal{G}} |\gamma_\sigma| \le 1 \quad \text{and} \quad \min_{\sigma \in \mathcal{G}} \gamma_\sigma \ge 0. \end{aligned} \tag{13}$$

The normalizing factor $|\mathcal{G}|^{-1}$ divides the ℓ_1-norm of γ by the total number of interaction patterns and is certainly not crucial here but simplifies notation. The estimation of $\hat{\gamma}$ is an application of Breiman's nonnegative Garrote (Breiman, 1995). As for the Garrote, the original predictors $\hat{T}_\sigma$ are not rescaled, thus putting effectively more penalty on unimportant predictors, with little variance of $\hat{T}_\sigma$ across the samples $X_1, \ldots, X_n$ and less penalty on the important predictors with higher variance, see (Yuan and Lin, 2007) for details.

The entire Forest Garrote algorithm works thus as follows

Forest Garrote

---

1. Fit Random Forest or another tree ensemble approach to the data.
2. Extract $\hat{T}_\sigma$ from the tree ensemble for all $\sigma \in \mathcal{G}$ by first extracting all rules R_j and corresponding regression coefficients $\hat{\beta}_j$ and grouping them via (10) for each interaction pattern.
3. Estimate $\hat{\gamma}$ as in (13) from the data, using for example the LARS algorithm (Efron *et al.*, 2004).
4. The fitted Forest Garrote function $\hat{T}^{gar}$ is given by (12).

The whole algorithm is very simple and fast, as there is no tuning parameter to choose.

Algorithmically, the problem can be solved with an efficient Lars algorithm (Efron *et al.*, 2004), which can easily be adapted to include the positivity constraint (Lykou and Whittaker, 2009). Alternatively, quadratic programming can be used.

It might be surprising to see the ℓ_1-norm constrained by 1 instead of a tuning parameter λ. Yet this is indeed one of the interesting properties of Forest Garrote. The tree ensemble is in some sense selecting a good level of sparsity. It seems maybe implausible that this would work in practice, but some intuitive reasons for its empirical success are given further below and ample empirical evidence is provided in the section with numerical results.

A drawback of the Garrote in the linear model setting is the reliance on the OLS estimator (or another suitable estimator), see also Yuan and Lin (2007). The OLS estimator is for example not available if $p > n$. Tree ensemble estimates are the initial estimators for Forest Garrote and they are, in contrast to OLS estimators, very reasonable estimators in a wide variety of settings, certainly including the high-dimensional setting $p \gg n$.

2.5. Lack of essential tuning parameter

In most regularization problems, like the Lasso (Tibshirani, 1996), choosing the regularization parameter is very important and it is usually a priori not clear what a good choice will be, unless the noise level is known with good accuracy (and it usually it is not). The most obvious approach would be cross-validation. Cross-validation can be computationally expensive and is usually not guaranteed to lead to optimal sparsity of the solution, selecting many more variables or interaction patterns than necessary, as shown for the Lasso in Meinshausen and Bühlmann (2006) and Leng, Lin and Wahba (2006).

The starting point for the Forest Garrote estimator is the tree ensemble. The tree ensemble has typically some tuning parameters. In the case of Random Forest primarily the minimal nodesize and the size of the random sets of variables across which the optimal splitpoint is chosen for each node. It is widely known, however, that Random Forests works close to optimal with the default tuning parameters. Increasing the minimal nodesize is mostly useful to speed up computation, especially for large datasets. (In the default setting, trees are grown until each node contains only very few observations and each tree contains thus on the order of n or $n \log n$ nodes.) Increasing the nodesize can also be helpful if the signal-to-noise ratio is very poor since it effectively decreases the variance of the estimator while inflating its bias. Changing the size of the random sets can improve performance but the improvements are mostly marginal. We assume that the default values are used in the Random Forest fit, even though it might be interesting to explore the possibility of further improvements by allowing some flexibility here.

Since Forest Garrote is starting from a tree ensemble, there is a natural tuning parameter in (13). As noted before, $\gamma = (1, 1, 1, \ldots, 1, 1)$ corresponds to the original tree ensemble solution. The original tree ensemble solution $\hat{T}$ is thus contained in the feasible region of the optimization problem (13), if the ℓ_1-norm is constrained to be less or equal to 1. The solution (12) will thus be at least as sparse as the original tree ensemble solution in the sense if sparsity is measured in the same ℓ_1-norm sense as in (13). Some variables might not be used at all even though they appear in the original tree ensemble.

On the other hand, the empirical squared error loss of $\hat{T}^{gar}$ is at least as low (and typically lower) than for the original tree ensemble solution as $\hat{\gamma}$ will reduce the empirical loss among all solutions in the feasible region of (13), which contains the original tree ensemble solution. The latter point does clearly not guarantee better generalization on yet unseen data, but a constraint of 1 on the ℓ_1-norm turns out to be an interesting starting point and is a very good default choice of the penalty parameter.

Sometimes one might still be interested in introducing a tuning parameter. One could replace the constraint of 1 on the ℓ_1-norm of $\gamma = (\gamma_{\sigma_1}, \ldots, \gamma_{\sigma_{|\mathcal{G}|}})$ in (13) by a constraint λ,

$$\begin{aligned} \hat{\gamma} \quad = \quad & \text{argmin}_{\gamma \in \mathbb{R}^{|\mathcal{G}|}} \sum_{i=1}^{n} \left(Y_i - \sum_{\sigma \in \mathcal{G}} \gamma_\sigma \hat{T}_\sigma(X_i) \right)^2 \\ & \text{such that} \quad |\mathcal{G}|^{-1} \sum_{\sigma \in \mathcal{G}} |\gamma_\sigma| \leq \lambda \quad \text{and} \quad \min_{\sigma \in \mathcal{G}} \gamma_\sigma \geq 0. \end{aligned} \tag{14}$$

The range over which to search, with cross-validation, to over λ can then typically be limited to $[0, 2]$. Empirically, it turned out that the default choice $\lambda = 1$ is very reasonable and actually achieves often better predictive and selection performance than the cross-validated solution, since the latter suffers from possibly high variance for finite sample sizes.

2.6. Example: diabetes data

The method is illustrated on the diabetes data from Efron *et al.* (2004) with $p = 10$ predictor variables, `age`, `sex`, body mass index (`bmi`), average blood pressure (`map`) and six blood serum measurements called `tc`, `ldl`, `hdl`, `tch`, `ltg`, `glu`, each scaled marginally to mean 0 and identical variance. These variables were obtained for each of $n = 442$ diabetes patients, along with the response of interest, a 'quantitative measure of disease progression one year after baseline'. Using Random Forest, the main effects and second-order interactions effects, extracted as in (10), are shown in the upper right diagonal of Figure 2, for 6 out of the 10 variables (chosen at random to facilitate presentation). All of these variables have non-vanishing main effects (on the diagonal) and the interaction patterns can be quite complex, making them somewhat difficult to interpret.

The way that Forest Garrote simplifies interaction terms is shown for the example of the interaction between the blood serum measurements `tch` and `ltg`. The leftmost columns shows the interaction between the two variables as it occurs in the Random Forest fit. The second column shows the decomposition into the four types of interaction proposed here. In the third column the effect of the Garrote estimation can be observed: three of the four interactions are shrunken to 0 and the overall interaction between these two variables is simplified considerably and made more amenable for interpretation.

Applying a Forest Garrote selection to the whole Random Forest fit, one obtains the main effects and interaction plots shown in the lower left diagonal

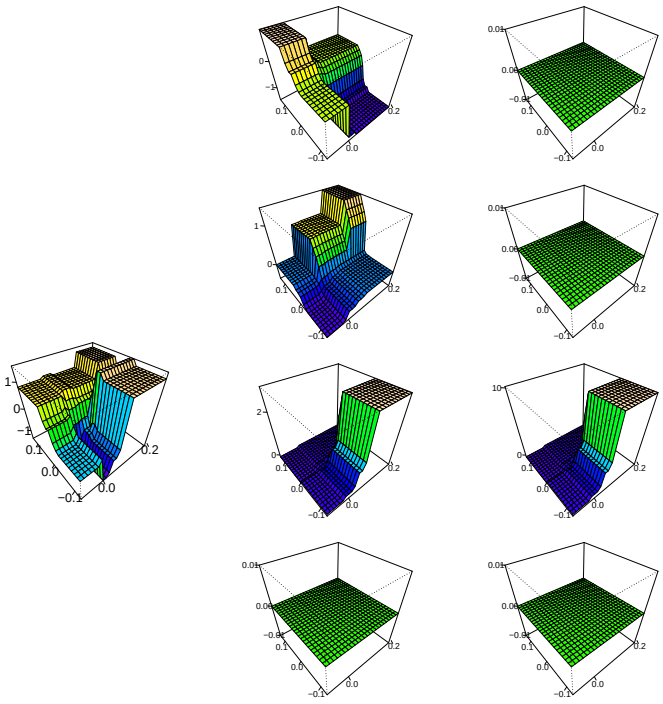

FIG 1. *FIRST COLUMN: combined variable interaction rules between blood serum concentrations* `tch` *(x-axis to the right) and* `ltg` *(y-axis to the left) in the Random Forest fit for the Diabetes data. SECOND COLUMN: the interaction rules can be decomposed into the effects* $\hat{T}_\sigma$ *(plotted on z-axis) of four interaction patterns* $\sigma = (+,-)$ *on top,* $(+,+)$ *on second from top,* $(-,+)$ *on second from bottom and* $(-,-)$ *on the bottom. THIRD COLUMN: applying the Garrote correction, three of the four interaction patterns are set to 0.*

of Figure 2. Note that the x-axis in the interaction plots corresponds to the variable in the same column, while the y-axis refers to the variable in the same row. Interaction plots are thus rotated by 90 degrees between the upper right and the lower left diagonal. Some main effects and interactions are set to 0 by the Forest Garrote selection. Interaction effects that are not set to 0 are typically 'simplified' considerably. The interaction plot of Forest Garrote seems thus much more amenable for interpretation.

3. Numerical results

To examine the predictive accuracy, variable selection properties and computational speed, various standard datasets are used and augmented with two higher-dimensional datasets. The first of these is a motif regression dataset (henceforth called 'motif', $p = 660$ and $n = 2587$). The goal of the data collection is to help find transcription factor binding sites (motifs) in DNA sequences. The

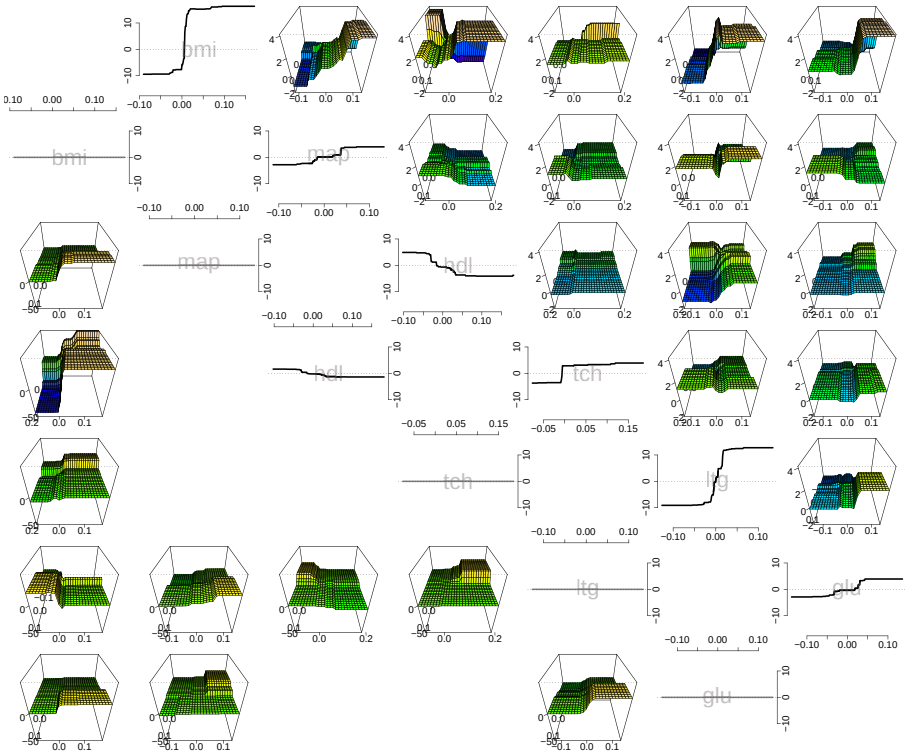

FIG 2. *UPPER RIGHT DIAGONAL: 'main effects' and 'interactions' of second degree for the Random Forest fit on the Diabetes data between the main 6 variables (not showing all variables). LOWER LEFT DIAGONAL: corresponding functions for the Forest Garrote. Some main effects and interactions are set exactly to zero. Vanishing interactions are not plotted, leaving some entries blank.*

real-valued predictor variables are abundance scores for p candidate motifs (for each of the genes). Our dataset is from a heat-shock experiment with yeast. For a general description and motivation about motif regression see Conlon *et al.* (2003).

The method is applied to a gene expression dataset ('vitamin') which is kindly provided by DSM Nutritional Products (Switzerland). For $n = 115$ samples, there is a continuous response variable measuring the logarithm of riboflavin (vitamin B2) production rate of Bacillus Subtilis, and there are $p = 4088$ continuous covariates measuring the logarithm of gene expressions from essentially the whole genome of Bacillus Subtilis. Certain mutations of genes are thought to lead to higher vitamin concentrations and the challenge is to identify those relevant genes via regression, possibly using also interaction between genes.

In addition, the diabetes data from Efron *et al.* (2004) ('diabetes', $p = 10, n = 442$), mentioned already above, are considered, the LA Ozone data ('ozone', $p = 9, n = 330$), and also the dataset about marketing ('marketing', $p =$

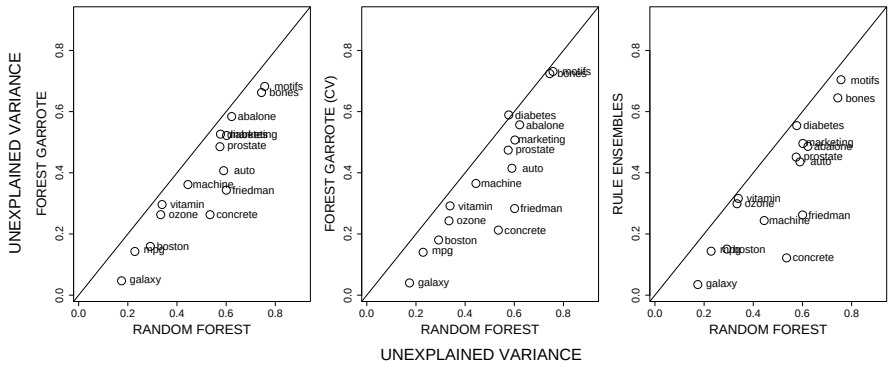

FIG 3. *The unexplained variance on test data, as a fraction of the total variance. LEFT: Comparison of unexplained variance for Forest Garrote versus Random Forests. MIDDLE: Forest Garrote (CV) versus Random Forests. RIGHT: Rule Ensembles versus Random Forests.*

$14, n = 8993$), bone mineral density ('bone', $p = 4, n = 485$), radial velocity of galaxies ('galaxies', $p = 4, n = 323$) and prostate cancer analysis ('prostate', $p = 9, n = 97$); the latter all from Hastie *et al.* (2001). The chosen response variable is obvious in each dataset. See the very worthwhile book Hastie *et al.* (2001) for more details. To give comparison on more widely used datasets, Forest Garrote is applied to various dataset from the UCI machine learning repository (Asuncion and Newman, 2007), about predicting fuel efficiency ('auto-mpg', $p = 8, n = 398$), compressive strength of concrete ('concrete', $p = 9, n = 1030$), median house prices in the Boston area ('housing', $p = 13, n = 506$), CPU performance ('machine', $p = 10, n = 209$) and finally the first of three artificial datasets in Friedman (1991).

The unexplained variance on test data for all these datasets with Forest Garrote is compared with that of Random Forests and Rule Ensembles in Figure 3. The tuning parameters in Random Forests (namely over how many randomly selected variables to search for the best splitpoint) is optimized for each dataset, using the out-of-bag performance measure. For comparison between the methods, the data are split into two parts of equal size, one half for training and the other for testing. Results are compared with Forest Garrote (CV), where the tuning parameter λ in (14) is not chosen to be 1 as for the standard Forest Garrote estimator, but is instead chosen by cross-validation. There are two main conclusions from the Figure. First, all three methods (Forest Garrote, Forest Garrote (CV) and Rule Ensembles) outperformed Random Forests in terms of predictive accuracy on almost all datasets. Second, the relative difference between these three methods is very small. Maybe surprisingly, using a cross-validated choice of λ did not help much in improving predictive accuracy for the Forest Garrote estimator. On the contrary, it rather lead to worse predictive performance, presumably due to the inherent variability of the selected penalty parameter.

TABLE 1

CPU time spent on Forest Garrote, Forest Garrote (CV) and Rule Ensembles for the various datasets. Time is given in seconds but results are only thought to be indicative of relative differences in computational requirements. Forest Garrote uses the least computational resources since (i) it starts from a relative small set of dictionary elements (all $\hat{T}_\sigma$ for $\sigma \in \mathcal{G}$ as opposed to all rules), (ii) the solution has to be computed only for a single regularization parameter and there is hence (iii) also no need for expensive cross-validation. Note that the times above are only for the rule selection steps (6) and (13) respectively and the overall relative speed difference is typically smaller since the same initial tree ensemble estimator is required in all settings.

dataset	n	p	Forest Garrote	Forest Garrote (CV)	Rule Ensembles
motifs	1294	666	88.5	425	933
ozone	165	9	0.26	2.7	16.5
marketing	3438	13	6.09	32.5	696
bones	242	3	0.05	1.1	26.2
galaxy	162	4	0.07	1.1	40.2
boston	253	13	0.79	5.3	87.5
prostate	48	8	0.25	3.3	2.4
vitamin	58	4088	1.98	21.6	39
diabetes	221	10	0.64	4.8	31.2
friedman	150	4	0.1	1.3	16.6
abalone	2088	8	0.59	5.1	808
mpg	196	7	0.18	1.9	36.6
auto	80	24	2.99	22	154
machine	104	7	0.29	2.5	19.6
concrete	515	8	0.58	4.3	271

Forest Garrote (CV) has also obviously a computational disadvantage compared with the recommended Forest Garrote estimator, as shown in Table 1 which is comparing relative CPU times necessary to compute the relative estimators. All three methods could be speeded up considerably by clever computational implementation. Any such improvement would most likely be applicable to any of these three compared methods as they have very similar optimization problems at their heart. Only relative performance measurements seem to be appropriate and only the time it takes to solve the respective optimization problems (6), (13) and (14) is reported, including time necessary for cross-validation, if so required. Rule Ensembles is faring by far the worst here, since the underlying optimization problem is very high-dimensional. The dimensionality J in (6) is the total number of rules, which corresponds to the number of all nodes in the Random Forest fit. The total number $|\mathcal{G}|$ of interaction patterns in the optimization underlying (13) in the Forest Garrote fit is, on the other hand, very much smaller than the number J of all rules, since many rules are typically combined in each interaction patterns. The lack of cross-validation for the Forest Garrote estimator clearly also speeds computation up by an additional factor between 5 and 10, depending on which form of cross-validation is employed.

Finally, the number of variables selected by either method is examined. A variable is said to be selected for this purpose if it appears in any node in the Forest or in any rule that is selected with a non-zero coefficient. In other words, selected variables will be needed to compute predictions, not selected variables can be discarded. The results are shown in Table 2. Many variables

TABLE 2
The number of variables selected in total for Forest Garrote, Rule Ensembles and Random Forests. Forest Garrote and Rule Ensembles prune the number of variables used considerably, especially for higher-dimensional data.

dataset	n	p	Forest Garrote	Rule Ensembles	Random Forests
motifs	1294	666	44	75	233
ozone	165	9	8	8	9
marketing	3438	13	9	13	13
bones	242	3	3	3	3
galaxy	162	4	4	4	4
boston	253	13	9	13	13
prostate	48	8	8	8	8
vitamin	58	4088	45	67	648
diabetes	221	10	7	8	10
friedman	150	4	4	4	4
abalone	2088	8	5	7	8
mpg	196	7	7	7	7
auto	80	24	16	15	21
machine	104	7	7	7	7
concrete	515	8	8	8	8

are typically involved in a Random Forest Fit and both Rule Ensembles as well as Forest Garrote can cut down this number substantially. Especially for higher-dimensional data with large number p of variables, the effect can be pronounced. Between Rule Ensembles and Forest Garrote, the differences are very minor with a slight tendency of Forest Garrote to produce sparser results.

4. Conclusions

Balancing interpretability and predictive power for regression problems is a difficult act. Linear models lend themselves more easily to interpretation but suffer often in terms of predictive power. Random Forests (RF), on the other hand, are known to deliver very accurate predictions. Tools exist to extract marginal variable importance measures from RF. However, the interpretability of RF could be improved if the very large number of nodes in the hundreds of trees fitted for RF could be reduced.

Here, the Forest Garrote was proposed as such a pruning method for RF or tree ensembles in general. It collects all rules or nodes in the Forest that belong to the same functional group. Using a Garrote-style penalty, some of these functional groups are then shrunken to zero, while the signal of other functional groups is enhanced. This leads to a sparser model and rather interpretable interaction plots between variables. Predictive power is similar or better to the original RF fit for all examined datasets.

The unique feature of Forest Garrote is that it seems to work very well without the use of a tuning parameter, as shown on multiple well known and less well known datasets. The lack of a tuning parameter makes the method very easy to implement and computationally efficient.

References

Asuncion, A. and Newman, D. (2007). UCI Machine Learning Repository. Available at http://www.ics.uci.edu/~mlearn/MLRepository.html

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics* **37** 373–384. MR1365720

Breiman, L. (2001). Random Forests. *Machine Learning* **45** 5–32.

Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth, Belmont. MR0726392

Chen, S., Donoho, S. and Saunders, M. (2001). Atomic Decomposition by Basis Pursuit. *SIAM Review* **43** 129–159. MR1854649

Conlon, E., Liu, X., Lieb, J. and Liu, J. (2003). Integrating regulatory motif discovery and genome-wide expression analysis. *Proceedings of the National Academy of Science* **100** 3339–3344.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least Angle Regression. *Annals of Statistics* **32** 407–451. MR2060166

Friedman, J. (1991). Multivariate adaptive regression splines. *Annals of Statistics* **19** 1–67. MR1091842

Friedman, J. and Popescu, B. (2008). Predictive learning via rule ensembles. *Annals of Applied Statistics* **2** 916–954.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J. and Tibshirani, R. (2001). *The Elements of Statistical Learning*. Springer New York. MR1851606

Leng, C., Lin, Y. and Wahba, G. (2006). A note on the lasso and related procedures in model selection. *Statistica Sinica* **16** 1273–1284. MR2327490

Lykou, A. and Whittaker, J. (2009). Sparse CCA using a Lasso with positivity constraints. *Computational Statistics & Data Analysis, to appear*.

Meier, L., van de Geer, S. and Buhlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society, Series B* **70** 53–71. MR2412631

Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the Lasso. *Annals of Statistics* **34** 1436–1462. MR2278363

Nash, W., Sellers, T., Talbot, S., Cawthorn, A. and Ford, W. (1994). The Population Biology of Abalone in Tasmania Technical Report, Sea Fisheries Division.

Strobl, C., Boulesteix, A., Zeileis, A. and Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics* **8** 25.

Strobl, C., Boulesteix, A., Kneib, T., Augustin, T. and Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics* **9** 307.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, Series B* **58** 267–288. MR1379242

Yu, B. and Bühlmann, P. (2003). Boosting with the L2 loss: Regression and classification. *Journal of the American Statistical Association* **98** 324–339. MR1995709

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B* **68** 49–67. MR2212574

Yuan, M. and Lin, Y. (2007). On the Nonnegative Garrote Estimator. *Journal of the Royal Statistical Society, Series B* **69** 143–161. MR2325269

Zhao, P., Rocha, G. and Yu, B. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *Annals of Statistics* **37** 3468–3497.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* **101** 1418–1429. MR2279469