The Annals of Applied Statistics
2010, Vol. 4, No. 4, 2049–2072
DOI: 10.1214/10-AOAS367

NODE HARVEST

BY NICOLAI MEINSHAUSEN

University of Oxford

When choosing a suitable technique for regression and classification with multivariate predictor variables, one is often faced with a tradeoff between interpretability and high predictive accuracy. To give a classical example, classification and regression trees are easy to understand and interpret. Tree ensembles like Random Forests provide usually more accurate predictions. Yet tree ensembles are also more difficult to analyze than single trees and are often criticized, perhaps unfairly, as 'black box' predictors.

Node harvest is trying to reconcile the two aims of interpretability and predictive accuracy by combining positive aspects of trees and tree ensembles. Results are very sparse and interpretable and predictive accuracy is extremely competitive, especially for low signal-to-noise data. The procedure is simple: an initial set of a few thousand nodes is generated randomly. If a new observation falls into just a single node, its prediction is the mean response of all training observation within this node, identical to a tree-like prediction. A new observation falls typically into several nodes and its prediction is then the weighted average of the mean responses across all these nodes. The only role of *node harvest* is to 'pick' the right nodes from the initial large ensemble of nodes by choosing node weights, which amounts in the proposed algorithm to a quadratic programming problem with linear inequality constraints. The solution is sparse in the sense that only very few nodes are selected with a nonzero weight. This sparsity is not explicitly enforced. Maybe surprisingly, it is not necessary to select a tuning parameter for optimal predictive accuracy. *Node harvest* can handle mixed data and missing values and is shown to be simple to interpret and competitive in predictive accuracy on a variety of data sets.

1. Introduction. Let $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_n)$ be a vector of n observations of a univariate real-valued response and $\mathbf{X}$ be the $n \times p$-dimensional matrix, where the row-vector $\mathbf{X}_{i\cdot} \in \mathcal{X}$ is the p-dimensional covariate for the ith observation for $i = 1, \ldots, n$. When trying to predict a new response, given covariates, regression trees [Breiman et al. (1984)] are attractive since they are very simple to build and understand. They are one example of a wider range of recursive partitioning methods. For the sake of notational simplicity, let the notion of a node in a tree and the corresponding subspace of $\mathcal{X}$ be identical. Let $\mathcal{Q}$ be a collection of q nodes, where a node $Q_g \in \mathcal{Q}$, $g = 1, \ldots, q$, is defined by a rectangular subspace of $\mathcal{X}$,

$$Q_g = \{\mathbf{x} \in \mathcal{X} : \mathbf{x}_k \in I_k^{(g)} \text{ for } k = 1, \ldots, p\},$$

Received October 2009; revised April 2010.

Key words and phrases. Trees, tree ensembles, machine learning, Random Forests, sparsity, quadratic programming.

and each interval $I_k^{(g)}$ is a subset of the support of the kth covariate.

The leaf nodes of a tree form a partition of $\mathcal{X}$ in that their union is identical to $\mathcal{X}$ and all pairwise intersections are empty. If each leaf node is an element of $\mathcal{Q}$, the partition corresponding to a tree can be expressed by a weight vector $\mathbf{w} \in \{0, 1\}^q$, where $\mathbf{w}_g = 0$ means that node g *is not* used in the partition, while $\mathbf{w}_g = 1$ means that node g *is* used in the partition. The tree-style prediction $\hat{Y}(\mathbf{x})$ at a point $\mathbf{x} \in \mathcal{X}$ is then the observed mean over all training observations in the same node,

$$\hat{Y}(\mathbf{x}) = \sum_{g=1}^{q} \mu_g 1\{\mathbf{x} \in Q_g\} \mathbf{w}_g, \tag{1}$$

where μ_g is the mean over all observations falling into node Q_g,

$$\mu_g = \frac{\sum_{i=1}^{n} 1\{\mathbf{X}_{i\cdot} \in Q_g\} \mathbf{Y}_i}{\sum_{i=1}^{n} 1\{\mathbf{X}_{i\cdot} \in Q_g\}}.$$

The predictions on the n observed samples can be conveniently written as $\mathbf{Mw}$, where $\mathbf{M}$ is the $n \times q$-dimensional matrix, with row entries for $i = 1, \ldots, n$ given by

$$\mathbf{M}_{ig} = \begin{cases} \mu_g, & \text{if } \mathbf{X}_{i\cdot} \in Q_g \\ 0, & \text{if } \mathbf{X}_{i\cdot} \notin Q_g \end{cases} \qquad \text{for } g = 1, \ldots, q = |\mathcal{Q}|. \tag{2}$$

The empirical squared error loss on the training samples is then

$$\|\mathbf{Y} - \mathbf{Mw}\|^2 \tag{3}$$

and trees try to pick a partitioning by a tree (and a weight vector $\mathbf{w}$ equivalently) that minimizes this empirical loss (3), under certain complexity constraints on the tree. These complexity constraints can, for example, entail a penalty on tree size or a lower bound on the number of observations in each node [Breiman et al. (1984)]; for an alternative approach see Blanchard et al. (2007). The optimal values of the complexity constraints are typically determined by cross-validation.

Compared to single regression trees, predictive accuracy is often improved by tree ensembles. Boosting [Freund and Schapire (1996); Friedman, Hastie and Tibshirani (2000)], bagging [Breiman (1996a)] and Random Forests [Breiman (2001)] are popular techniques to create these ensembles. Predictions are weighted averages over the output of all trees in the ensemble. They thus effectively allow an observation to be part of more than one node. For Random Forests [Breiman (2001)], each of m trees in the ensemble receives equal weight $1/m$. If all leaf nodes of the Random Forest are part of the set $\mathcal{Q}$ above, the empirical loss can again be written as in (3) with the only difference that now $\mathbf{w}_g \in \{0, 1/m, 2/m, \ldots, 1\}$ instead of the binary weights $\mathbf{w}_g = \{0, 1\}$ for trees. If a node appears only once in the ensemble, its weight is $1/m$. If it appears more than once, the associated weight is the corresponding multiple of $1/m$, up to a maximum of 1 if the node appears in every tree of the ensemble.

Here, we explore the possibility of allowing arbitrary weights $\mathbf{w}_g \in [0, 1]$. Rather than growing trees greedily, we start from a large set $\mathcal{Q}$ of potential nodes that are either obtained by random splits or picked from an initial tree ensemble, just as in 'Rule ensembles' [Friedman and Popescu (2008)]. While 'Rule ensembles' uses the nodes as binary indicator variable in a linear model with an ℓ_1-penalty on coefficients, *node harvest* retains tree-like predictions of the form (1). The only task of *node harvest* is finding suitable weights on nodes. Minimizing the empirical loss (3) under suitable constraints on the weights turns out to be a quadratic program with linear inequality constraints, which can be solved efficiently.

The goal of the proposed *node harvest* procedure is two-fold: On the one hand, a very competitive predictive accuracy (with practically no adjustment of tuning parameters). On the other hand, simple, interpretable results and predictions.

Random Forests satisfy the first of these demands but not necessarily the latter since hundreds of large trees with thousands of nodes are involved in the final decision. Marginal importance measures can be calculated as proposed in [Breiman (2001)], but they only describe some limited characteristics of the fitted function and certainly do not explain the whole fit. Trees, on the other hand, satisfy the second constraint but fall short of optimal predictive accuracy. Moreover, if tree size is chosen by cross-validation, the interaction order (tree depth) can be very high, lowering interpretability. *Node harvest* has the advantage of delivering very accurate results while using in general only main effects and two-factor interactions.

Node harvest is introduced in Section 2. An extension to binary classification, dealing with missing values and additional regularization of the estimator are covered in Section 3, while numerical results are shown in Section 4.

2. Node harvest.

Node harvest (NH) is introduced, along with an efficient algorithm to solve the involved quadratic programming problem. Some basic properties of the estimator are established.

2.1. *Optimal partitioning*.

The starting point of NH is loss function (3). Suppose one would like to obtain a partitioning of the space that minimizes the empirical loss (3). One could collect a very large number of nodes into the set $\mathcal{Q}$ that satisfy desired complexity criteria. Typical complexity criteria are a minimal node size or maximal interaction depth (tree depth). An empirically optimal partitioning would search for a weight vector such that the empirical loss is minimal,

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{Y} - \mathbf{M}\mathbf{w}\|^2 \quad \text{such that } \mathbf{w} \in \{0, 1\}^q \text{ and } \{Q_g : \mathbf{w}_g = 1\} \text{ is a partition of } \mathcal{X}. \tag{4}$$

The selected set $\{Q_g : \mathbf{w}_g = 1\} \subset \mathcal{Q}$ of nodes is understood to form a partition iff the intersection between all selected nodes is empty and their union is the entire space $\mathcal{X}$. Even if given a collection $\mathcal{Q}$ of nodes, the optimization problem above

is very difficult to solve. The constraint $\mathbf{w} \in \{0, 1\}^q$ does not correspond to a convex feasible region. Moreover, the constraint that the selected set of nodes form a partition of the space is also awkward to handle computationally.

The latter problem can be circumvented by demanding instead that the partition is a proper partitioning for the *empirically observed data only* in the sense that each data point is supposed to be part of exactly one node. This loosening of the constraint will be very helpful at a later stage. It might create the situation that a new observation will not belong to any node, but this will turn out to be not a problem in the NH approach since every observation will be a member of the root node and the root node always receives a small positive weight, which is discussed further below.

To form such an empirical partitioning, let $\mathbf{I}$ be the $n \times q$ matrix indicating whether or not an observation falls into a given leaf. For all rows $i = 1, \ldots, n$,

$$\mathbf{I}_{ig} = \begin{cases} 1, & \text{if } \mathbf{X}_{i\cdot} \in Q_g \\ 0, & \text{if } \mathbf{X}_{i\cdot} \notin Q_g \end{cases} \qquad \text{for } g = 1, \ldots, q. \tag{5}$$

The constraint that each data point be part of one and exactly one node is equivalent to demanding that $\mathbf{Iw} = 1$, understood componentwise. Since $\mathbf{w} \in \{0, 1\}^q$, this simple linear equality constraint ensures that each observation is part of exactly one selected node.

Given a collection $\mathcal{Q}$ of nodes, a weight vector $\mathbf{w}$ could thus be found by the constrained optimization

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{Y} - \mathbf{Mw}\|^2 \qquad \text{such that } \mathbf{Iw} = 1 \text{ and } \mathbf{w} \in \{0, 1\}^q. \tag{6}$$

For the n observed data points, this problem is equivalent to (4), yet it is still NP-hard to solve in general due to the nonconvex feasible region of the constraint $\mathbf{w} \in \{0, 1\}^q$. Tree ensembles relax this constraint and average over several trees, implicitly allowing weights to take on values in the interval $[0, 1]$. It thus seems natural to relax the nonconvex constraint $\mathbf{w} \in \{0, 1\}^q$ and only ask for nonnegativity of the weights.

2.2. *Node harvest.* The main idea of NH is that it becomes computationally feasible to solve the optimal empirical partitioning problem (6) if the weights are only constrained to be nonnegative. The weights across all nodes for a single observation still have to sum to 1 (as they do for all weighted tree ensembles), but this constraint is equivalent to $\mathbf{Iw} = 1$, and we can relax (6) to the convex optimization problem

$$\hat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{Y} - \mathbf{Mw}\|^2 \qquad \text{such that } \mathbf{Iw} = 1 \text{ and } \mathbf{w} \geq 0. \tag{7}$$

This estimator is called the *node harvest* (NH) estimator since a small subset of nodes is 'picked' or selected from a large initial ensemble of nodes. It will turn out that the vast majority of nodes in this large ensemble will receive a zero weight,

without the sparsity being enforced explicitly other than through the constraint $\mathbf{I}\mathbf{w} = 1$. Nodes g which receive a zero weight ($\hat{\mathbf{w}}_g = 0$) can be ignored for further analysis.

The constraints in (7) are satisfied, for example, by setting the weight of the root node, which is always included in $\mathcal{Q}$ and contains all observations, equal to 1 and all other weights to 0. The set of solutions is thus always nonempty. The solution to (7) is also either unique or the set of solutions is a convex set. In the latter case, we define $\hat{\mathbf{w}}$ for definiteness to be the solution that has minimal ℓ_2-norm among all solutions in this convex set, which amounts to adding a small ridge penalty $\nu\|\mathbf{w}\|_2^2$ to the objective function in (7) and letting $\nu \to 0$. Other solutions are possible, but adding a very small ridge penalty guarantees, moreover, positive definiteness of the quadratic form and facilitates computation of (7) even if the solution is unique.

The prediction for new data is then simply a weighted average over node means. For the training data, this is still the vector $\mathbf{M}\mathbf{w}$. The prediction $\hat{Y}(\mathbf{x})$ for a new data point $\mathbf{x} \in \mathcal{X}$ is the weighted average over all nodes that $\mathbf{x}$ falls into,

$$\hat{Y}(\mathbf{x}) = \frac{\sum_{g \in G_{\mathbf{x}}} \hat{\mathbf{w}}_g \mu_g}{\sum_{g \in G_{\mathbf{x}}} \hat{\mathbf{w}}_g}, \tag{8}$$

where $G_{\mathbf{x}} := \{g : \mathbf{x} \in Q_g\}$ is the collection of nodes that observation $\mathbf{x}$ falls into.

The denominator in (8) is constrained to be 1 for all n training samples since $\mathbf{I}\mathbf{w} = 1$ is enforced. For new observations outside the training set, the weights in the denominator do not necessarily sum to 1. We always let the root node be a member of the set $\mathcal{Q}$, where the root node is defined as containing the entire predictor space $\mathcal{X}$. We demand that the weight of the root node is bounded from below not by 0 as for all other nodes, but by a very small weight chosen here as 0.001 and converging to 0 for increasing sample sizes. The set $G_{\mathbf{x}}$ in (8) is then always nonempty and the denominator in (8) is bounded from below by 0.001, although it will typically be in the region of 1 for new observations. In the unlikely event that a new observation is not part of any node except the root node, its prediction will, according to (8), be the node mean of the root node. This is identical to the mean response over all observations in the training data, a reasonable answer if a new observation should fail to fall into any selected node.

2.3. *Tuning parameters*. The NH procedure requires only an initial set of nodes $\mathcal{Q}$. Once this set is specified, there are no further tuning parameters. It will turn out that results are very insensitive to the actual choice of the set of nodes as long as $q = |\mathcal{Q}|$ is sufficiently large and some complexity constraints, such as maximal interaction order and minimal node size, are followed.

There are three essential characteristics of the set $\mathcal{Q}$: the number of nodes, maximal interaction order and minimal node size. We discuss these constraints in the following, but an advantageous aspect of the proposed method is that the method is competitive in terms of predictive accuracy for the default choices proposed below. In fact, all numerical results are computed with the same defaults parameters

for maximal interaction order, which is set to 1, and minimal node size, which is set to 5.

Number of nodes. It will be shown empirically for many data sets that the performance is continuously improving the more nodes $q = |\mathcal{Q}|$ are added to the initial set of nodes. Solving (7) gets clearly more costly as q increases. One should thus use as many nodes as can be afforded computationally. Typically, q ranges in the hundreds or thousands. All examples are calculated with $q = 1000$ nodes. It is maybe surprising that there is practically no overfitting if q is chosen very large. A first attempt at explaining this phenomenon can be found in Proposition 1.

Maximal interaction order. The maximal interaction order of node Q_g is the number of variables that are necessary to determine whether an observation is part of a node or not. Main effects have thus an interaction order 1. To keep results as interpretable as possible, a maximal interaction order of 2 (equivalent to a two-factor interaction) is chosen for almost all examples.

Minimal node size. The minimal node size $\min_g |\{i : \mathbf{X}_{i\cdot} \in Q_g\}|$ has an influence on the amount of smoothing. Allowing nodes with just a single observation, the algorithm could simply interpolate all observed data by assigning weights of 1 to the n nodes that contain each exactly one of the n observations. This is clearly undesirable and a minimal node size of 5 is imposed throughout. Again, results could be improved for some data sets by tuning this choice, yet the results show that a choice of 5 gives very competitive results across a remarkably wide range of data sets.

2.4. *Node generation*. To generate the desired nodes, one can generate nodes at random, without use of the response variable. Alternatively, one can use a data-adaptive choice by using nodes from a fitted tree ensemble. Results seem very insensitive to this choice, but the latter method requires in general fewer nodes in the initial set $\mathcal{Q}$ for a close to optimal predictive accuracy. We thus follow the latter approach. The set $\mathcal{Q}$ is initially empty. A new tree is grown as proposed in Breiman (2001) for each tree in a Random Forest (RF) ensemble. To speed up computation and increase diversity of the set, the trees are fitted on subsamples of the data of size $n/10$ rather than bootstrap samples. All the nodes of the tree that satisfy the maximal interaction order and minimal node size constraint are added to the set $\mathcal{Q}$, provided that they are not already present in the set. While the size of $\mathcal{Q}$ is less than the desired number q, the procedure is repeated. If two or more nodes in $\mathcal{Q}$ contain exactly the same set of training observations, only a randomly chosen one of them is kept.

2.5. *Algorithm and dimensionality reduction*. As stated above, the initial set of nodes $\mathcal{Q}$ is generated with a Random Forests approach. After the desired number q of nodes have been obtained, it only remains to solve (7). This is a quadratic program (QP) with linear constraints and could be solved with standard QP solvers.

However, the specific structure of the problem can be used to reduce dimensionality and make the computation more efficient.

We suppose that the root node, containing all observations, is the first among all $q = |\mathcal{Q}|$ nodes. Let $\mathbf{w}_{\text{root}}$ be the vector $\mathbf{w}_{\text{root}} = (1, 0, 0, \ldots, 0)$. Clearly, $\mathbf{I}\mathbf{w}_{\text{root}} = 1$ componentwise, so the equality constraint in (7) is fulfilled for $\mathbf{w}_{\text{root}}$. This means that the difference $\hat{\mathbf{w}} - \mathbf{w}_{\text{root}}$ between the actual solution and the 'root' solution $\mathbf{w}_{\text{root}}$ lies in the nullspace $\mathcal{N}_{\mathbf{I}} \subseteq \mathbb{R}^q$ of $\mathbf{I}$. Let $\tilde{q}$ be the dimension of $\mathcal{N}_{\mathbf{I}}$. Since $\mathbf{I}$ is of rank at most $\min\{q, n\}$, we have $\tilde{q} \geq q - \min\{q, n\}$, and the nullspace $\mathcal{N}_{\mathbf{I}}$ is guaranteed to be nontrivial ($\tilde{q} > 0$) for $q > n$, that is, if there are more nodes than actual observations, which we can always satisfy by generating sufficiently many nodes. If the nullspace is nontrivial, then let $\mathbf{B}$ be the $q \times \tilde{q}$-dimensional matrix, where the kth column, with $k = 1, \ldots, \tilde{q}$, contains the kth basis vector of an arbitrarily chosen orthonormal basis of $\mathcal{N}_{\mathbf{I}}$. The solution to (7) can then be written, using the argument above, for some $\hat{\mathbf{d}} \in \mathbb{R}^{\tilde{q}}$ as $\hat{\mathbf{w}} = \mathbf{w}_{\text{root}} + \mathbf{B}\hat{\mathbf{d}}$, and, to get the same solution as in (7), $\hat{\mathbf{d}}$ is the solution to

$$\hat{\mathbf{d}} = \operatorname*{argmin}_{\mathbf{d}} -2\mathbf{d}^T(\mathbf{MB})^T(\mathbf{Y} - \overline{\mathbf{Y}}) + \mathbf{d}^T(\mathbf{MB})^T(\mathbf{MB})\mathbf{d} \quad \text{such that } \mathbf{Bd} \geq -\mathbf{w}_{\text{root}}, \tag{9}$$

where it was used that $\mathbf{M}\mathbf{w}_{\text{root}} = \overline{\mathbf{Y}}$ by definition of $\mathbf{w}_{\text{root}}$. If a small ridge penalty $\nu\|\mathbf{w}\|_2^2$ on $\mathbf{w}$ is added to guarantee uniqueness of the solution, a term $\nu\|(\mathbf{w}_{\text{root}} + \mathbf{Bd})\|_2^2$ is added to the objective function in (9), where here always $\nu = 0.001$ under a standardized response with $\text{Var}(\mathbf{Y}) = 1$. To also ensure that the weight of the root node is bounded from below by the small chosen value 0.001 instead of 0, the constraint $\mathbf{Bd} \geq -\mathbf{w}_{\text{root}}$ in (9) needs to be replaced by $\mathbf{Bd} \geq -0.999\mathbf{w}_{\text{root}}$.

Thus, the original q-dimensional problem is reduced to a $\tilde{q} \geq q - \min\{q, n\}$-dimensional one. A price to pay for this is the computation of a basis for the nullspace $\mathcal{N}_{\mathbf{I}}$ of $\mathbf{I}$, which is achieved by a SVD of $\mathbf{I}$. Compared to the savings in the QP solution, computation of the SVD is, however, very much worthwhile. The remaining QP problem (9) is solved with the QP solver of Goldfarb and Idnani (1983), as implemented in the package `quadprog` of the `R`-programming language [R Development Core Team (2005)]. It is conceivable that an alternative interior-point algorithm and especially explicit use of the sparse structure of the matrixes $\mathbf{M}$ and $\mathbf{I}$ would generate additional computational savings, but, even so, it took less than 10 seconds to solve (9) on data sets with less than 10^3 observations, using a 2.93 GHz processor and 8 GB of RAM.

2.6. *Smoothing*. NH can be seen as a smoothing operation in that $\hat{\mathbf{Y}} = \mathbf{SY}$ for a data-adaptive choice of the smoothing matrix $\mathbf{S}$. The smoothing matrix is doubly stochastic, symmetric and has nonnegative entries.

LEMMA 1. *The fitted values* $\hat{\mathbf{Y}}$ *are obtained as a linear transformation* $\hat{\mathbf{Y}} = \mathbf{SY}$ *of the original data, where* $\mathbf{S}$ *is a doubly stochastic and symmetric matrix in*

that $\sum_j \mathbf{S}_{ij} = 1$ *for all* $i = 1, \ldots, n$ *and* $\sum_i \mathbf{S}_{ij} = 1$ *for all* $j = 1, \ldots, n$. *Moreover,* $\mathbf{S}_{ij} \geq 0$ *for all* $i, j = 1, \ldots, n$.

PROOF. The fitted values are for the n training observations given by $\hat{\mathbf{Y}} = \mathbf{M}\hat{\mathbf{w}}$, with $\mathbf{M}$ defined in (2). Therefore, $\hat{\mathbf{Y}}_i = \sum_{g=1}^{q} 1\{i \in Q_g\}\hat{\mathbf{w}}_g \mu_g$, where $i \in Q_g$ is a shorthand notation for $\mathbf{X}_{i\cdot} \in Q_g$. Let $n_g = |\{j : j \in Q_g\}|$ be the number of samples in node g. Then $\mu_g = n_g^{-1} \sum_{j \in Q_g} \mathbf{Y}_j$ by definition of the node means and, hence, putting together,

$$\hat{\mathbf{Y}}_i = \sum_{g=1}^{q} 1\{i \in Q_g\}\hat{\mathbf{w}}_g n_g^{-1} \sum_{j=1}^{n} 1\{j \in Q_g\}\mathbf{Y}_j = \sum_{j=1}^{n} \sum_{g=1}^{q} \frac{\hat{\mathbf{w}}_g 1\{i, j \in Q_g\}}{n_g} \mathbf{Y}_j.$$

Defining matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ by its entries $\mathbf{S}_{ij} = \sum_g \hat{\mathbf{w}}_g n_g^{-1} 1\{i, j \in Q_g\}$, it follows that (a) $\hat{\mathbf{Y}} = \mathbf{S}\mathbf{Y}$, (b) $\mathbf{S}$ is symmetric and (c) that all entries are nonnegative. It remains to show that $\sum_j \mathbf{S}_{ij} = 1$ for all $i = 1, \ldots, n$. The column sums follow by symmetry. Now, $\sum_j \mathbf{S}_{ij} = \sum_j \sum_g \hat{\mathbf{w}}_g n_g^{-1} 1\{i, j \in Q_g\} = \sum_g \hat{\mathbf{w}}_g 1\{i \in Q_g\}$. By definition of the matrix $\mathbf{I}$, the right-hand side $\sum_g \hat{\mathbf{w}}_g 1\{i \in Q_g\}$ is identical to the ith coefficient in $\mathbf{I}\hat{\mathbf{w}}$. Since, componentwise, $\mathbf{I}\hat{\mathbf{w}} = 1$ by (7), it follows that indeed $\sum_j \mathbf{S}_{ij} = 1$ for all $i = 1, \ldots, n$, which completes the proof. □

From the lemma above, one can immediately derive that the mean $\overline{\hat{\mathbf{Y}}}$ of the fitted values is identical to the mean $\overline{\mathbf{Y}}$ of the observed values. And the lemma above also ensures that, irrespective of the size q of the initial ensemble, it is impossible to fit the response exactly by interpolation if the minimal node size is strictly larger than 1.

PROPOSITION 1. *The mean of the fitted and observed values agree,* $\overline{\hat{\mathbf{Y}}} = \overline{\mathbf{Y}}$. *Moreover, if the minimal node size is larger than* 1, *the weight of the root node is strictly positive and* $\mathrm{Var}(\mathbf{Y}) \neq 0$, *it holds for any strictly convex real-valued function* f *that*

$$\sum_{i=1}^{n} f(\hat{\mathbf{Y}}_i) < \sum_{i=1}^{n} f(\mathbf{Y}_i). \tag{10}$$

PROOF. The first claim follows directly from Lemma 1 since $\hat{\mathbf{Y}} = \mathbf{S}\mathbf{Y}$ and, hence, $\sum_{i=1}^{n} \hat{\mathbf{Y}}_i = \sum_{i,j=1}^{n} \mathbf{S}_{ij}\mathbf{Y}_j = \sum_{j=1}^{n} \mathbf{Y}_j$, where the last equality follows by the fact that $\sum_i \mathbf{S}_{ij} = 1$ for all $j = 1, \ldots, n$ from Lemma 1. Likewise, observe that $\mathbf{S}_{ij} < 1$ for all $i, j = 1, \ldots, n$ if the minimal node size is larger than 1. This follows from the definition of $\mathbf{S}$ by the entries $\mathbf{S}_{ij} = \sum_g \hat{\mathbf{w}}_g n_g^{-1} 1\{i, j \in Q_g\}$ since more than 1 entry in each row-vector $\mathbf{S}_{i\cdot}$, $i = 1, \ldots, n$, has to be nonzero. Since the sum of the row is constrained to $\sum_j \mathbf{S}_{ij} = 1$ and all entries in $\mathbf{S}$ are nonnegative, all entries have got to be strictly less than 1. Moreover, if the weight of the root

node is positive, all entries $\mathbf{S}_{i,j}$ are strictly positive. Hence, for a strictly convex function f,

$$\sum_{i=1}^{n} f(\hat{\mathbf{Y}}_i) = \sum_{i=1}^{n} f\left(\sum_{j=1}^{n} \mathbf{S}_{ij}\mathbf{Y}_j\right) < \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{S}_{ij} f(\mathbf{Y}_j) = \sum_{j=1}^{n} f(\mathbf{Y}_j),$$

having used $\mathrm{Var}(\mathbf{Y}) \neq 0$ and the strict positivity of all entries of $\mathbf{S}$ in the inequality and $\sum_j \mathbf{S}_{ij} = 1$ for all $i = 1, \ldots, n$, from Lemma 1 in the last equality. □

The second part of the result can be obtained if the condition that the weight of the root nodes is positive is replaced with the following weaker condition: there exists a pair of observations $\mathbf{Y}_i, \mathbf{Y}_j$ with $\mathbf{Y}_i \neq \mathbf{Y}_j$ such that both i and j are members of a node $\mathcal{Q}_g$ and the weight $\hat{\mathbf{w}}_g$ is strictly positive.

The proposition implies that the observed data cannot be interpolated exactly by NH even though the number q of nodes might greatly exceed sample size n.

2.7. *Related work.* There has been substantial interest in the Random Forests framework for classification and regression [Breiman (2001)], which builds partly upon the randomized tree idea in Amit and Geman (1997). Lin and Jeon (2006) interpreted Random Forests as an adaptive nearest neighbor scheme, with the distance metric given by the grown tree ensemble. The same interpretation is maybe even more imminent for NH since predictions are explicitly averages over node means. Both bagging [Breiman (1996a)] and boosting [Freund and Schapire (1996); Friedman, Hastie and Tibshirani (2000)] are possible alternative and powerful techniques for growing multiple trees. If using either of these, predictions are formed by averaging in a possibly weighted form across all grown trees. Results are often difficult to interpret, though, as each of possibly hundreds of grown trees consists in turn of multiple nodes and all variables in the data set are often involved somewhere in the ensemble. The influence of individual variables can only be measured indirectly for such tree ensembles; see Strobl et al. (2007) for a more involved discussion. Despite a similar sounding name, 'tree harvesting' [Hastie et al. (2001)], a regression technique commonly used in computational biology, is not closely related to NH. An interesting machine learning technique is 'stacking' [Wolpert (1992); Breiman (1996b)], which is weighting various classifiers and weights are chosen by minimizing the error on weighted leave-one-out predictions. In contrast to stacked trees, however, NH is not weighting whole trees but is working at the level of individual nodes by reweighting each node. In a similar spirit, the 'Rule Ensemble' algorithm by Friedman and Popescu (2008) simplifies interpretability of tree ensembles by selecting just a few nodes across all trees. Each node is seen to form a binary indicator variable and the prediction is a linear combination of all these indicator variables. In fact, for a given collection $\mathcal{Q}$ of nodes, the matrix whose columns correspond to the binary indicator variables is exactly the matrix defined as $\mathbf{I}$ in (5). The linear combination β of nodes is then sought in

a Lasso-style way by putting a constraint on the ℓ_1-norm of the coefficient vector [Tibshirani (1996); Chen, Donoho and Saunders (2001)],

$$\hat{\beta}^{\lambda} = \underset{\beta}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{I}\beta\|^2 \qquad \text{such that } \|\beta\|_1 \leq \lambda. \tag{11}$$

The original variables can be added to the matrix $\mathbf{I}$ of binary indicator variables. Despite the superficial similarity of 'Rule Ensembles' with NH, there are fundamental differences to the NH procedure (7). Choosing the right tuning parameter λ is essential in (11), but no such tuning is necessary for NH. The inherent reason for this is that NH imposes much stronger regularization by requiring in (8) that predictions are weighted node means. NH is only selecting the weights $\mathbf{w}$ in (7), whereas the vector β in (11) cannot be interpreted as the weight attached to a particular node or rule. The sign and magnitude of the coefficient β_g is thus not directly related to the average response of observations in node g. A possible advantage of NH is thus the interpretability of the predictions as weighted node means. An example is shown in the breast cancer example in Figure 3. If a new patient falls into only a single node, the NH prediction is simply the average response in the group of patients, which is very easy to communicate and relate to the actually observed data. If he or she falls into several groups, the prediction is the weighted average across these groups. In terms of predictive power, rule ensembles seem to be often better than NH and also Random Forests in our experience if the signal-to-noise is high [Meinshausen (2009)]. The strength of NH is its ability to cope well with very low signal-to-noise ratio data and the two approaches seem complementary in this regard. Both 'Rule Ensembles' and NH can make use of a dictionary of rules, which is currently built either randomly or by harvesting nodes from existing tree ensembles such as Random Forests. More general nodes, such as spheres under various metrics that are centered at training observations, could conceivably help improve both methods.

3. Extensions. *Node harvest* (NH) can be extended and generalized in various ways, as briefly outlined below. NH is shown to be directly applicable to binary classification. Missing values can easily be dealt with, without using imputation techniques or surrogate splits when predicting the response for new observations with missing values. Finally, a regularization is proposed that can reduce the number of selected nodes.

3.1. *Classification.* For binary classification with $Y \in \{0, 1\}$, the nonconvex misclassification loss is typically replaced with a convex majorant of this loss function [Bartlett, Jordan and McAuliffe (2003)]. One of these possible convex loss functions is the L_2-loss, as used for classification in Yu and Bühlmann (2003).

Simply applying the previous QP problem (7) on binary data leads to a prediction $\hat{Y}(\mathbf{x})$ at a new data point $\mathbf{x}$ which is identical to (8). The node means μ_g,

$g = 1, \ldots, q$, are now equivalent to the fraction of samples in class "1" among all samples in node Q_g,

$$\mu_g = \frac{|\{i : \mathbf{X}_{i\cdot} \in Q_g \text{ and } \mathbf{Y}_i = 1\}|}{|\{i : \mathbf{X}_{i\cdot} \in Q_g\}|}.$$

The NH predictions are naturally in the interval $[0, 1]$. Use of the L_2-loss as a convex surrogate for misclassification error is thus not only appropriate for NH, it is even beneficial since it allows for an interpretation of the predictions $\hat{Y}(\mathbf{x})$ as weighted empirical node means.

3.2. *Missing values*. An interesting property of NH is its natural ability to cope with missing values. Once a fit is obtained, predictions for new data can be obtained without use of imputation techniques or surrogate splits. To fit the *node harvest* estimator with missing data, we replace missing values in the matrix $\mathbf{X}$ by the imputation technique described in Breiman (2001) and Liaw and Wiener (2002) and proceed just as previously.

Suppose then that the *node harvest* estimator is available and one would like to get a prediction for a new observation $\mathbf{X}_{i\cdot}$ that has missing values in some variables. We still calculate the prediction as the weighted mean (8) over all nodes of which the new observation is a member. The question is whether observation i is part of node $Q_g \in \mathcal{Q}$ if it has missing values in variables that are necessary to evaluate group membership of node Q_g. The simplest and, as it turns out, effective solution is to say that i is not a member of a node if it has missing values in variables that are necessary to evaluate membership of this node. To make this more precise, let Q_g be a node

$$Q_g = \{\mathbf{x} \in \mathcal{X} : \mathbf{x}_k \in I_k^{(g)} \text{ for all } k \in \{1, \ldots, p\}\},$$

and let $\mathcal{K}_g \subseteq \{1, \ldots, p\}$ be the set of variables that are necessary and sufficient to evaluate node membership [sufficient in the sense that $I_k^{(g)}$ is identical to the entire support of $\mathbf{x}_k$ for all $k \notin \mathcal{K}_g$ and necessary in the sense that $I_k^{(g)}$ is *not* identical to the support of $\mathbf{x}_k$ for all $k \in \mathcal{K}_g$]. If $\mathbf{x}$ has missing values, we define

$$\mathbf{x} \in Q_g \text{ if and only if, for all } k \in \mathcal{K}_g, \mathbf{x}_k \text{ is not missing and } \mathbf{x}_k \in I_k^{(g)}.$$

Since we usually only work with main effects and two-factor interactions, all nodes require only one or two variables to evaluate node membership. Even with missing values in $\mathbf{X}_{i,\cdot}$, observation i can still be a member of many nodes in $\mathcal{Q}$, namely, those that involve only variables where the ith observation has nonmissing values. In the most extreme case, *all* variables are missing from a new observation. The observation will then *only* be a member of the root node and the prediction is the node mean of the root node, which is the mean of the response variable across all training observations, maybe not an unreasonable answer in the absence of any

information. In more realistic cases, the new observation will have *some* nonmissing variables and be a member of more than the root node and the prediction will be more refined. With trees, a similar idea would amount to dropping a new observation down a tree and stopping at the first node where the split-variable is missing. The prediction would then naturally be the mean response of observations within this node. However, if the variables on which the root node is split are missing, the predicted response will be the mean across all observations. This situation occurs for NH only typically if all variables are missing. The use of surrogate variables [Breiman et al. (1984)] is thus paramount for trees, while NH can take a more direct approach.

3.3. *Regularization*. There is so far no tuning parameter in the NH procedure apart from the choice of the large initial set $\mathcal{Q}$ of nodes. And results are rather insensitive to the choice of $\mathcal{Q}$ as long as it is chosen large enough, as shown in the next section with numerical results.

Even though often not necessary from the point of predictive accuracy, the method can be regularized to further improve interpretability. Here it is proposed to constrain the average number of samples in each node. From the outset, the minimal node size of 5 ensures that the average fraction of samples in each node is above $5/n$. Even so, one might not like to select many nodes that contain only a handful of observations. The fraction of samples in node g is n_g/n and the weighted mean across all nodes is

$$\frac{\sum_g \hat{\mathbf{w}}_g (n_g/n)}{\sum \hat{\mathbf{w}}_g}, \tag{12}$$

where $n_g = |\{j : j \in Q_g\}|$ is again the number of samples in node g. Since $\mathbf{I}\hat{\mathbf{w}} = 1$ by (7), we have, by summing over the rows of this equality,

$$n = \sum_{i=1}^{n} \sum_{g=1}^{q} \mathbf{I}_{ig} \hat{\mathbf{w}}_g = \sum_{g=1}^{q} \hat{\mathbf{w}}_g \sum_{i=1}^{n} \mathbf{I}_{ig} = \sum_{g=1}^{q} \hat{\mathbf{w}}_g n_g,$$

where the last equality stems from the definition of matrix $\mathbf{I}$ in (5). The nominator in (12) is thus 1 and the weighted average fraction of samples (12) within nodes is, maybe surprisingly, equal to the inverse of the ℓ_1-norm of the weight vector $\hat{\mathbf{w}}$,

$$\frac{\sum_g \hat{\mathbf{w}}_g (n_g/n)}{\sum_g \hat{\mathbf{w}}_g} = \frac{1}{\sum_g \hat{\mathbf{w}}_g} = \|\hat{\mathbf{w}}\|_1^{-1}. \tag{13}$$

Constraining the ℓ_1-norm of $\hat{\mathbf{w}}$ to be less than a positive value of $\lambda \in [1, \infty]$ constrains thus the average fraction of samples (13) to be at least $1/\lambda$. For $\lambda = 1$, every node with nonzero weight has to contain all n samples and only the root node is thus selected for $\lambda = 1$. At the other extreme, let m be the minimal node size (here $m = 5$). For $\lambda > n/m$, the constraint will have no effect at all, since all nodes

have $n_g \geq m$ anyhow and the average weighted fraction (13) is thus bounded from below by m/n for all weight vectors. The regularized estimator $\hat{\mathbf{w}}^\lambda$ solves then

$$\hat{\mathbf{w}}^\lambda = \operatorname*{argmin}_{\mathbf{w}} \|\mathbf{Y} - \mathbf{M}\mathbf{w}\|^2 \quad \text{such that } \mathbf{I}\mathbf{w} = 1 \text{ and } \mathbf{w} \geq 0 \text{ and } \|\mathbf{w}\|_1 \leq \lambda \tag{14}$$

instead of (7). The interesting region is $\lambda \in [1, m/n]$, where m is the enforced lower bound on node size. From the point of predictive accuracy, constraining λ is usually not beneficial unless the signal-to-noise ratio is very low. There is thus a tradeoff between sparsity (number of selected nodes) and predictive power, as shown in the next section with numerical results.

4. Numerical results. For various data sets, we look at the predictive accuracy of *node harvest* (NH) and various related aspects like sensitivity to the size of the initial set of nodes, interpretability and predictive power of results under additional regularization as in (14).

4.1. *Example I*: *Two-dimensional sinusoidal reconstruction*. As a very simple first example, assume that the random predictor variable $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$ is two-dimensional and distributed uniformly on $[0, 1]^2$. and the response is generated as

$$Y = \sin(2\pi \mathbf{x}_1) \sin(2\pi \mathbf{x}_2) + \varepsilon, \tag{15}$$

where ε follows a normal distribution with mean 0 and variance $1/4$ and the noise is independent between observations. Taking $n = 10^3$ samples from (15), a regression tree [Breiman et al. (1984)] is fitted to the data, using a cross-validated choice of tree size penalty. The fit is constant on rectangular regions of the two-dimensional space, as shown in Figure 1. Each of these regions corresponds to a node in the tree. The fit is rather poor, however, and the structure of the problem is not well captured. Random Forests is fitted with the default parameters proposed in Breiman (2001). It improves in terms of predictive accuracy on trees, yet the contour plot appears very noisy since the trees are grown until almost pure (keeping only 10 observations in each node) and the variability of the Random Forests approach manifests itself here in a high spatial variability of the fitted function. NH is fitted with the default parameters used throughout (1000 random nodes generated picked from a Random Forest fit, two-factor interactions and minimal node size of 10). It gives a comparably clean contour plot, as seen in the rightmost panel of Figure 1 and forms a compromise between trees and Random Forests. In contrast to trees, the fitted function is not constant across rectangular-shaped subspaces since each observation can fall into more than one node.

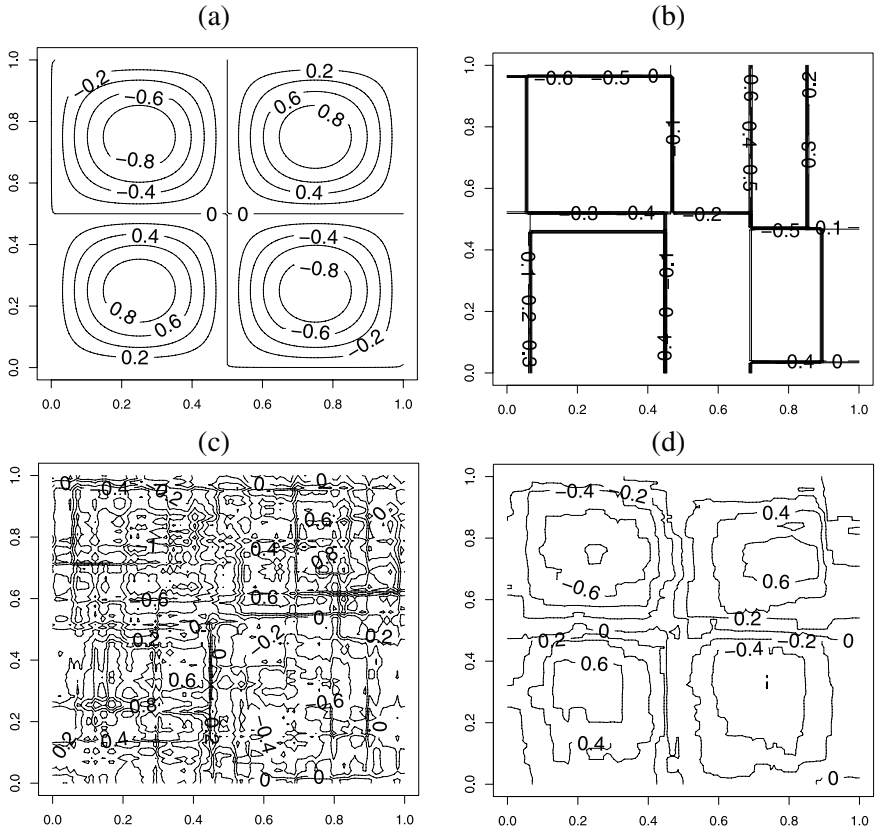

FIG. 1. (a) *Contour plot of* $E(Y)$ *under model* (15) *in the two-dimensional predictor space, with contour lines at values* -1 *to* 1 *with step sizes of* 0.2. *The contour plot for the fit of a regression tree* (b), *a Random Forest fit* (c) *and node harvest* (d). *The three methods are fitted using the same* 10^3 *observations from* (15).

4.2. *Example II*: *Importance sampling in climate modeling*. The *climateprediction.net* project [Allen (1999)] is, broadly speaking, concerned with uncertainty analysis of climate models, using a distributed computing environment. A climate model contains typically several parameters whose precise values are only known up to a certain precision. The project analyzes the behavior of a coarse resolution variant of the HadCM3 climate model [Johns et al. (2003)] under thousands of small perturbations of the default parameters. Once a certain number of models has been sampled, the behavior of the underlying climate model can be better understood and importance sampling can be used to sample only in relevant sections of the parameter space. While Gaussian process emulation is widely used in this context [Oakley and O'Hagan (2004)], we note that the data here are not noise free since the outcome depends on the random initial conditions and a standard regression analysis of the model is hence useful. Without giving a full explanation,

we show an example of a data set containing 250 models, each run with a different combination of 29 parameters. The response variable is mean temperature change over a 50 year period under a given emissions scenario.

Following the approach laid out above, 1000 nodes are generated with a Random Forest type approach. All of these nodes are constrained to contain at least 10 observations and have at most two-factor interactions. Then the quadratic program (7) is applied. Only 14 of the originally 1000 nodes receive a nonzero weight and these nodes are shown in Figure 2.

The plot is very interpretable: the position of each node on the x-axis corresponds to the mean of all training observations in this node. And predictions for new data are simply the weighted mean across all nodes the new observation falls into. The weight of each node is proportional to the area with which it is plotted.

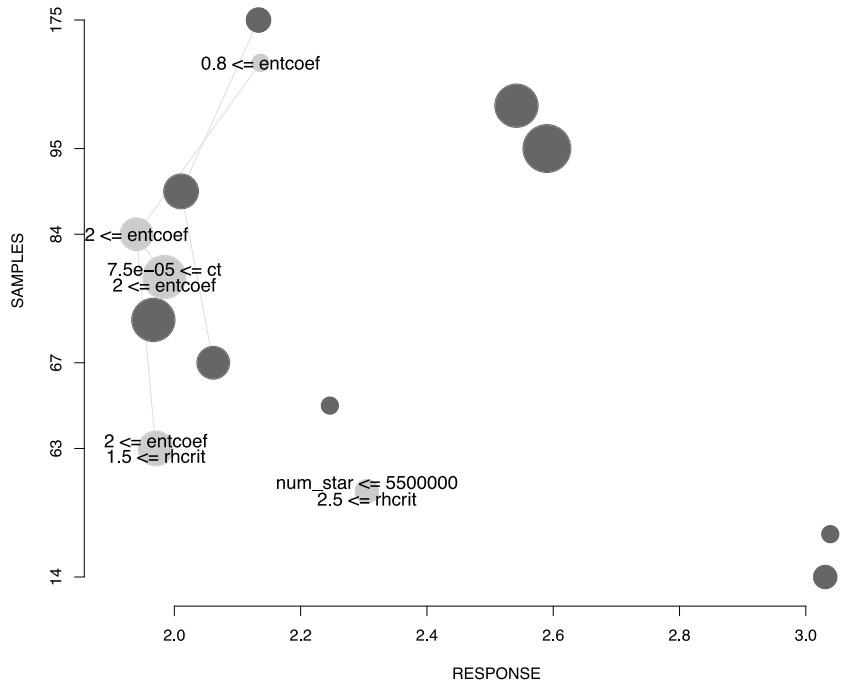

FIG. 2. *The* 14 *nodes selected by node harvest for the* climateprediction.net *data. The area of each node* g *is proportional to the weight* $\hat{\mathbf{w}}_g$ *it received in* (7). *The* 4986 *nodes that received a zero weight are not shown. The position on the x-axis shows for each node* g *the mean* μ_g *of all training observations that fall into it, while the position on the y-axis shows how many observations it contains. If observations of a node are a subset of observations of another node, a line between the two nodes is drawn. The node "entcoef* ≥ 2*" contains a subset of the observations of the node "entcoef* ≥ 0.8*." A single new observation was chosen at random and the* 5 *nodes that the new observation falls into are lighter and annotated. The prediction for the new observation is then simply the weighted mean across the x-axis positions of the annotated nodes.*

To give an example of a prediction, a new observation is sampled at random. It happens to fall into five nodes, whose respective weights and node means are as follows:

Node g	*entcoef* ≥ 2 *ct* $\geq 7.5 \cdot 10^{-5}$	*entcoef* ≥ 2 *rhcrit* ≥ 1.5	*entcoef* ≥ 2	*num_star* $\leq 5.5 \cdot 10^5$ *rhcrit* ≥ 2.5	*entcoef* $\geq .8$
Mean μ_g	1.98	1.97	1.94	2.30	2.14
Weight $\hat{\mathbf{w}}_g$	0.37	0.24	0.21	0.11	0.06

Four of these nodes contain the entrainment coefficient (*entcoef*) as a split variable, which is maybe unsurprising since the entrainment coefficient is known to be the parameter to which the model is most sensitive.

The new observation belongs also to the root node (as do all observations), with the minimal imposed weight 0.001 for this node, but this influence is negligible and ignored here. The predicted response for this new observation is then the weighted mean across these nodes, which is 2.014. A graphical visualization of this weighted averaging is immediate from Figure 2. The prediction for this new observation (or rather model) is simply the weighted horizontal position of the 5 selected and annotated nodes, with weights proportional to node size. As will be seen further below, the predictive accuracy of NH is for this data set better than cross-validated trees, even though no tuning was used in the NH approach and the result is at least as interpretable and simple as a tree. To get optimal predictive performance, a tree needs to employ interactions up to fourth order while NH gets a better accuracy with only two-factor interactions.

4.3. *Example III*: *Wisconsin breast cancer data*. As an example of binary classification, take the Wisconsin breast cancer data [Mangasarian, Street and Wolberg (1995)]. There are 10 clinical variables to predict whether a tumor is benign or malignant. Applying NH again with 1000 RF-generated nodes, with at most two-factor interactions and a minimal node size of 10, the results in Figure 3 are obtained. The root node is again not shown, despite its small enforced positive weight of 0.001. The position on the x-axis gives for each node the percentage of people within this group that had a malignant tumor ($Y = 1$). The y-axis position is proportional to the number of people within this node in the training sample. A new patient falls into one or several of these nodes and the predicted probability of class $Y = 1$ for this patient is simply the weighted average over the means μ_g of all nodes g the patient is part of, as shown for a randomly chosen example patient in Figure 3. A prediction (or risk assessment in the example) is thus easy to communicate and can be related to the empirical outcome in relevant groups of patients with similar characteristics.

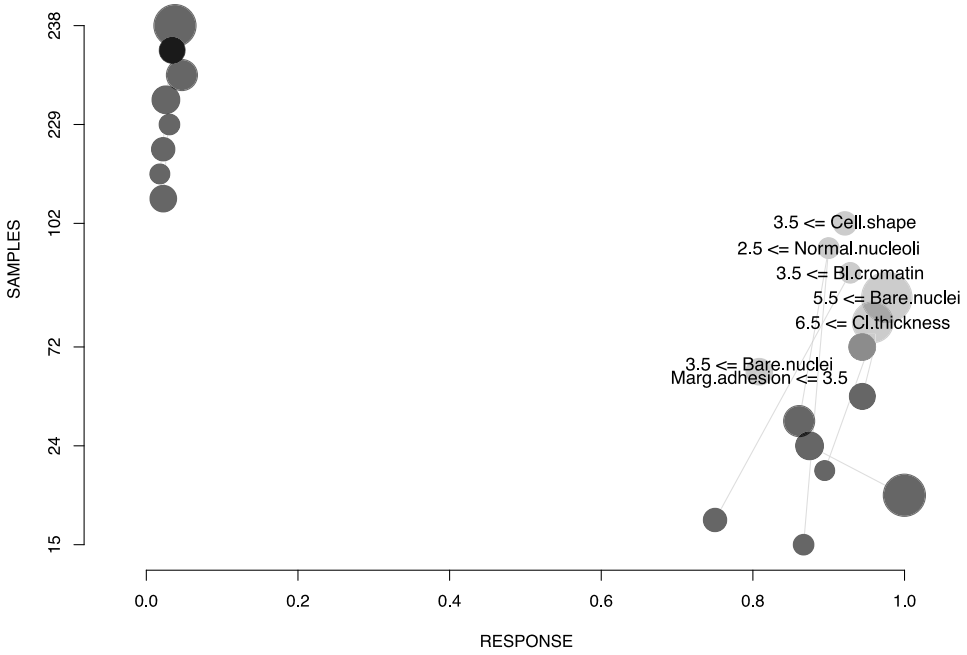

FIG. 3. *Node harvest (NH) estimator for the Wisconsin Breast Cancer study.* 22 *nodes are selected, where the number of patients within each node is shown on the vertical scale. The percentage of patients with a malignant tumor* ($Y = 1$) *is shown for each node on the horizontal scale. The number of patients within each node is shown on the vertical scale. The size of nodes is again plotted proportional to the weights chosen by NH. A new patient was randomly selected and belongs to the* 6 *lighter and annotated nodes. Among these, there are* 5 *'main effect' nodes, with the addition of one 'two-factor interaction' node. All of the* 6 *selected groups of patients contain a large fraction of people with a malignant tumor, with actual proportions varying between* 83% *for node "Bare.nuclei* ≥ 3.5; *Marg.adhesion* ≤ 3.5*" to above* 97% *for node "bare.nuclei* ≥ 5.5*." The estimated probability for having a malignant tumor for this new patient is the weighted mean across the percentages of people with a malignant tumor in these* 6 *groups of patients.*

If splitting the data into two equally large parts and taking one part as training and the other part as test data, and averaging over 20 splits, the misclassification test error with NH is 3.6%, compared with 3.3% for Random Forests and 5.5% for cross-validated classification trees. NH seems to perform better in a low signal-to-noise ratio setting. If changing 20% of all labels in the training set, the performance of Random Forests drops to 6.0% while NH maintains an accuracy of 4.4%. This behavior is completely analogous to regression, as shown below.

4.4. *Further data sets.* Besides these examples, the method is applied to motif regression [Conlon et al. (2003)], where the task is to identify transcription factor binding sites from gene expression measurements. The data set consist of $n = 2588$ samples and $p = 660$ genes and the response variable is the concentration of the transcription factor. In addition, the well-known abalone data

[Nash et al. (1994)], with $p = 8$, are considered, as are the diabetes data from Efron et al. (2004) ('diabetes,' $p = 10, n = 442$) and the LA Ozone data ('ozone,' $p = 9, n = 203$), bone mineral density data ('bones,' $p = 4, n = 485$), fuel efficiency data ('mpg,' $p = 7, n = 392$), median house prices in the Boston area ('housing,' $p = 13, n = 506$), CPU performance data ('machine,' $p = 7, n = 209$), crime rate data from the US census ('crime,' $p = 101, n = 1993$), and a data set about prediction of Parkinson's symptoms from voice measurements ('parkinson,' $p = 19, n = 5875$). The latter data sets are all available at the UCI machine learning repository [Asuncion and Newman (2007)]. We also consider a data set about radial velocity of galaxies ('galaxy,' $p = 4, n = 323$) and prostate cancer analysis ('prostate,' $p = 8, n = 97$), the latter all from Hastie, Friedman and Tibshirani (2001), which contains more details, and, finally, a gene expression data set, kindly provided by DSM nutritional products (Switzerland). For $n = 115$ samples, there is a continuous response variable measuring the logarithm of riboavin (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. Observations with missing values are removed from the data sets. Even though NH could deal with these, as alluded to above, it facilitates comparison with other techniques.

Each data set is split 10 times into two equally large parts. On the half used as a training set, NH is employed as well as Random Forests (RF), a CART regression tree (TREE), Rule Ensembles (RE) and L_2-boosted regression trees (L_2B). For NH, we select 1000 nodes from the Random Forest ensemble as described above, keeping only main-effect and two-factor interaction nodes and a minimal node size of 5. Then (7) is applied to this ensemble and exactly the same procedure is followed for all data sets without any tuning of these parameters. The same initial set of nodes is used for Rule Ensembles with a 5-fold CV-choice of the tuning parameter. We remark that both NH and RE could perform better for some data sets if higher order interactions were allowed in the nodes. For Random Forests, one could fine tune the minimal node size or the value of *mtry*, which is the size of the random number of variables used to find the optimal split point at each node. However, they are kept at the default values (which are known to give nearly optimal results), as proposed in Breiman (2001) and Liaw and Wiener (2002), to give an equal comparison between the two essentially 'tuning'-free algorithms NH and RF. The size of the regression trees [Breiman et al. (1984)] is chosen by 10-fold CV on the training data. Boosting is using regression trees of depth two as weak learners and a CV-optimized stopping time. The predictions on the test data (the second part of the data) are then recorded for all three methods and the fraction of the variance that is unexplained is averaged across all 10 sample splits. The number of training observations available for each data set is shown in Table 1, together with the average unexplained fraction of the variance.

TABLE 1

Average proportion of unexplained variance on test data, rounded to two significant figures for Random Forests (RF), CART trees (TREE), Node Harvest without regularization, $\lambda = \infty$, (NH_∞), Rule Ensembles (RE) and L_2-boosted regression trees (L_2B)

								With additional observational noise				
Data set	***n***	***p***	**RF**	**TREE**	**RE**	**L_2B**	**NH_∞**	**RF**	**TREE**	**RE**	**L_2B**	**NH_∞**
Ozone	203	12	**0.27**	*0.41*	0.31	0.33	0.34	0.55	>1	>1	0.67	**0.47**
Mpg	392	7	**0.15**	*0.24*	0.16	0.16	0.20	0.54	>1	0.47	0.39	**0.36**
Servo	166	4	0.32	0.38	**0.20**	0.37	0.26	0.61	*0.94*	0.73	0.88	**0.57**
Prostate	97	8	**0.53**	*0.68*	0.61	0.63	0.58	>1	>1	>1	>1	>1
Housing	506	13	**0.14**	*0.30*	0.18	0.19	0.25	0.46	>1	0.66	0.48	**0.39**
Diabetes	442	10	**0.55**	*0.71*	0.58	0.57	0.59	0.74	>1	>1	0.74	**0.65**
Machine	209	7	**0.16**	0.58	*0.86*	0.43	0.27	0.84	>1	>1	0.56	**0.54**
Galaxy	323	4	**0.036**	*0.094*	0.045	0.049	0.065	0.53	*0.81*	0.35	0.33	**0.26**
Abalone	4177	8	**0.45**	0.56	0.52	0.48	*0.60*	0.64	*0.65*	0.59	**0.56**	0.61
Bones	485	3	0.71	*0.79*	0.73	0.73	**0.70**	**0.83**	>1	0.98	0.88	0.85
Cpdn	493	29	**0.52**	0.66	0.55	*0.68*	0.66	0.98	>1	0.98	0.98	**0.77**
Motifs	2587	666	**0.67**	*0.87*	0.72	0.71	0.78	0.83	>1	>1	0.84	**0.80**
Vitamin	115	4088	**0.35**	*0.55*	0.40	0.38	0.37	**0.78**	>1	>1	0.99	0.98
Crime	1993	101	**0.34**	*0.47*	0.38	0.36	0.42	0.46	*0.70*	0.49	0.46	**0.45**
Parkinson	5875	19	**0.20**	0.33	0.53	0.63	*0.69*	**0.43**	0.68	0.60	*0.76*	0.69

Notes: The best performing method is shown in bold, while the worst performing method is shown in italics. A result '>1' indicates that the prediction is worse on test data than the best constant prediction.

On most data sets, Random Forests has the highest predictive accuracy with the exception of 'servo' and 'bones,' where NH is coming on top. A single tree is, maybe unsurprisingly, consistently the worst performing method. The picture changes if additional noise is added to the training observations. To this end, the response vector $\mathbf{Y}$ is replaced on the training observations with the response $\mathbf{Y} + \varepsilon$, where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)$ contains i.i.d. standard normal noise with variance three times the variance of $\mathbf{Y}$, cutting the correlation between the true unknown signal and the response exactly in half. As can be seen in the right part of the table, NH is now the best performing method on the clear majority of these low signal-to-noise ratio data, sometimes outperforming all other approaches by a substantial margin.

Figure 4 shows the impact that the number of nodes in the initial set $\mathcal{Q}$ has on predictive accuracy: the more nodes in $\mathcal{Q}$, the better the predictive accuracy on test data. Even though Figure 4 shows this phenomenon only up to a few thousands of nodes, it holds well beyond this point. In other words, NH does not seem to overfit if more and more nodes are added to the initial set of nodes and it is ideal to include as many nodes as computationally feasible in $\mathcal{Q}$, even though a few hundred seem to be sufficient for most data sets.

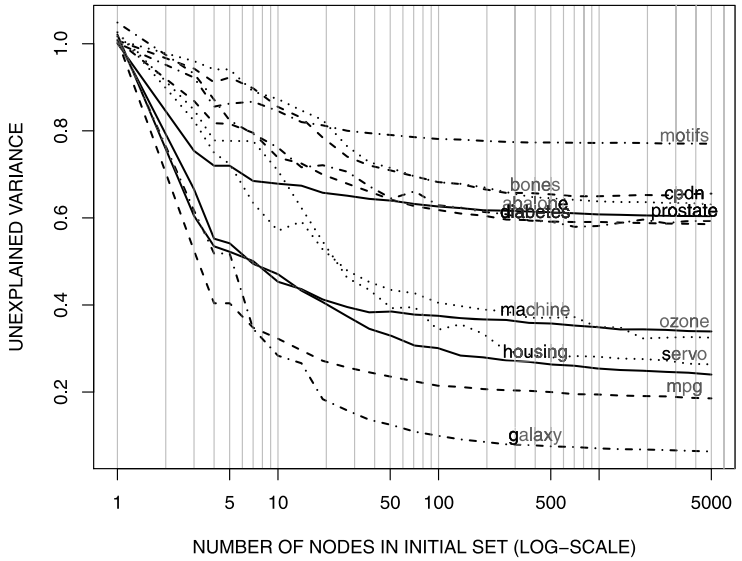

FIG. 4. *The unexplained variance on test data as a function of the number q of nodes in the initial set of nodes* (*x-axis in log-scale*). *Each line corresponds to one data set. Close to optimal performance is reached after a few hundred nodes, with results continuing to improve slightly thereafter.*

A crude measure for the complexity of a tree or tree ensembles is the total number of nodes of the predictor, which is equivalent to the total number of leaf nodes for tree ensembles and the total number of nodes with nonzero weights or coefficients for NH and RE respectively. Table 2 shows that NH (with $\lambda = \infty$) and RE use roughly a similar amount of nodes in the final fit, typically a few dozen, while NH with regularization yields the sparsest results in general, with the obvious exception of single trees, as seen in the following Table 3. Boosting leads to hundreds and Random Forests to thousands or even hundreds of thousands of final leaf nodes. The greater sparsity of NH and RE comes at a higher computational price. Starting from the same number of initial nodes, NE and RE are more computationally intensive to compute than all other methods, with a slight edge for NH, especially for data sets with a larger sample size. While it is faster to fit RF than either RE or NH, it should be emphasized that, due to much fewer used nodes, NH and RE are clearly very fast for predicting the response of new observations, which can be of importance in an online prediction setting, where RF can be too slow for some applications.

Last, the effect of regularization (14) on the sparsity of the solution and predictive accuracy is examined. Results are summarized in Table 3, where the unconstrained estimator is compared for all previous data sets with the regularized estimator at $\lambda = 3$. Unsurprisingly, regularization always improves the sparsity of the solution. The average number of selected nodes can decrease by a potentially substantial amount if applying the additional regularization, improving interpretabil-

TABLE 2

Average number of nodes for each tree-based predictor (left half) and the average computational time necessary to fit the predictor in seconds (right half), rounded to two significant figures

			Number of leaf nodes					Computational time (s)				
Data set	n	p	**RF**	**TREE**	**RE**	L_2**B**	**NH**$_\infty$	**RF**	**TREE**	**RE**	L_2**B**	**NH**$_\infty$
Ozone	203	12	$>10^4$	6.7	32	98	97	0.15	0.016	11	0.57	25
Mpg	392	7	$>10^4$	8	38	150	56	0.17	0.015	28	0.53	8.6
Servo	166	4	$>10^4$	2.9	21	110	20	0.074	0.0098	3.8	0.27	3.2
Prostate	97	8	$>10^4$	3.9	20	71	52	0.22	0.01	2.8	0.27	8.8
Housing	506	13	$>10^4$	8.3	60	130	74	0.36	0.025	84	0.53	24
Diabetes	442	10	$>10^4$	12	36	96	75	0.26	0.019	57	0.5	31
Machine	209	7	$>10^4$	3.6	62	420	47	0.24	0.011	9	0.34	6.7
Galaxy	323	4	$>10^4$	5.8	49	170	52	0.19	0.0098	19	0.35	8.6
Abalone	4177	8	$>10^4$	11	74	150	53	5.8	0.16	520	0.76	38
Bones	485	3	$>10^4$	10	27	67	30	0.15	0.011	20	0.45	4.6
Cpdn	493	29	$>10^4$	13	44	82	24	0.42	0.041	35	0.71	4.8
Motifs	2587	666	$>10^4$	15	68	120	64	140	11	470	100	86
Vitamin	115	4088	$>10^4$	4.7	43	230	70	2.5	0.92	4.6	170	75
Crime	1993	101	$>10^4$	11	59	140	71	12	0.83	220	3	21
Parkinson	5875	19	$>10^4$	24	97	100	15	16	0.53	920	1.4	44

ity. Predictive accuracy is typically very similar between the two estimators, with an advantage for the unconstrained estimator for the original data sets. Regularization seems to improve the already very good performance of NH in the low signal-to-noise ratio setting where additional noise is applied to the training data. Overall, the unconstrained estimator seems a very good default choice. Applying the additional regularization is worthwhile if the results are desired to be very sparse or the signal in the data is extremely weak.

5. Discussion. The aim of *node harvest* (NH) is to combine positive aspects of trees on the one hand and tree ensembles such as Random Forests on the other hand.

NH shares with trees the ease of interpretability and simplicity of results. As with trees, only a few nodes are used. For trees, every observation falls exactly into one such node and the predicted response is the corresponding node mean. With NH, nodes can overlap and an observation can be a member of a few nodes. While trees often have to include higher order interactions to achieve their optimal predictive performance, it is often sufficient for NH to include main effects and two-factor interactions. While tree size is determined by cross-validation, essentially no tuning parameter and no cross-validation is necessary for NH.

The lack of a very important tuning parameter is thus a common feature of both NH and Random Forests. Predictive accuracy also seems comparable. For

TABLE 3

Average proportion of unexplained variance and average number of selected nodes for the unrestricted node harvest estimator ($\lambda = \infty$) and the regularized estimator ($\lambda = 3$), where the average fraction of samples in each node has to be larger than $\lambda^{-1} = 1/3$. The better performing method is again shown in bold

					With additional noise			
	Unexpl. variance		**No. selected nodes**		**Unexpl. variance**		**No. selected nodes**	
Data set	$\lambda = \infty$	$\lambda = 3$	$\lambda = \infty$	$\lambda = 3$	$\lambda = \infty$	$\lambda = 3$	$\lambda = \infty$	$\lambda = 3$
Ozone	**0.34**	0.34	97	**73**	**0.47**	0.48	100	**95**
Mpg	**0.20**	0.24	56	**37**	0.36	**0.34**	35	**34**
Servo	**0.26**	0.27	20	**11**	0.57	**0.49**	23	**17**
Prostate	0.58	**0.57**	52	**47**	>1	**0.98**	53	**47**
Housing	**0.25**	0.28	74	**40**	**0.39**	0.41	69	**46**
Diabetes	**0.59**	0.61	75	**55**	**0.65**	0.66	96	**96**
Machine	0.27	**0.26**	47	**35**	0.54	**0.48**	42	**40**
Galaxy	**0.065**	0.097	52	**32**	0.26	**0.24**	44	**33**
Abalone	**0.60**	0.63	53	**38**	**0.61**	0.63	39	**28**
Bones	0.70	**0.70**	30	**22**	0.85	**0.83**	30	**24**
Cpdn	**0.66**	0.68	24	**18**	**0.77**	0.79	48	**36**
Motifs	**0.78**	0.78	64	**46**	0.80	**0.80**	54	**42**
Vitamin	**0.37**	0.39	70	**60**	1.00	**0.85**	71	**66**
Crime	**0.42**	0.44	71	**44**	**0.45**	0.48	59	**46**
Parkinson	**0.69**	0.71	15	**12**	**0.69**	0.73	22	**18**

high signal-to-noise ratio data, Random Forests seems to have an edge while NH delivers typically a smaller loss if the signal-to-noise ratio drops to lower values. The general advantage of NH over Random Forests is simplicity and arguably much better interpretability of results.

In common with both trees and tree ensembles, NH can handle mixed data very well and is invariant under monotone transformations of the data. NH is, moreover, able to deal with missing values without explicit use of imputation or surrogate splits. Both regression and classification are handled naturally and it is conceivable that the method can also be extended to censored data, in particular, survival analysis, in analogy to the extension of Random Forests to Random Survival Forests [Ishwaran et al. (2006)]. Most of the functionality of *node harvest* is implemented in the package `nodeHarvest` for the `R`-programming language [R Development Core Team (2005)].

Acknowledgments. I would like to thank a referee, an Associate Editor and the editor Michael Stein for their helpful comments on an earlier version of the manuscript.

REFERENCES

ALLEN, M. (1999). Do-it-yourself climate prediction. *Nature* **401** 642–642.

AMIT, Y. and GEMAN, D. (1997). Shape quantization and recognition with randomized trees. *Neural Comput.* **9** 1545–1588.

ASUNCION, A. and NEWMAN, D. (2007). *UCI Machine Learning Repository*. Univ. California, Irvine, CA.

BARTLETT, P., JORDAN, M. and MCAULIFFE, J. (2003). Convexity, classification, and risk bounds. Technical report, Dept. Statistics, UC Berkeley.

BLANCHARD, G., SCHÄFER, C., ROZENHOLC, Y. and MÜLLER, K. (2007). Optimal dyadic decision trees. *Mach. Learn.* **66** 209–241.

BREIMAN, L. (1996a). Bagging predictors. *Mach. Learn.* **24** 123–140.

BREIMAN, L. (1996b). Stacked regressions. *Mach. Learn.* **24** 49–64.

BREIMAN, L. (2001). Random forests. *Mach. Learn.* **45** 5–32.

BREIMAN, L., FRIEDMAN, J., OLSHEN, R. and STONE, C. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA. MR0726392

CHEN, S., DONOHO, S. and SAUNDERS, M. (2001). Atomic decomposition by basis pursuit. *SIAM Rev.* **43** 129–159. MR1854649

CONLON, E., LIU, X., LIEB, J. and LIU, J. (2003). Integrating regulatory motif discovery and genome-wide expression analysis. *Proc. Natl. Acad. Sci.* **100** 3339–3344.

EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004). Least angle regression. *Ann. Statist.* **32** 407–451. MR2060166

FREUND, Y. and SCHAPIRE, R. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference* 148–156. Morgan Kauffman, San Francisko, CA.

FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2000). Additive logistic regression: A statistical view of boosting. *Ann. Statist.* **28** 337–407. MR1790002

FRIEDMAN, J. and POPESCU, B. (2008). Predictive learning via rule ensembles. *Ann. Appl. Statist.* **2** 916–954. MR2522175

GOLDFARB, D. and IDNANI, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Math. Program.* **27** 1–33. MR0712108

HASTIE, T., FRIEDMAN, J. and TIBSHIRANI, R. (2001). *The Elements of Statistical Learning*. Springer, New York. MR1851606

HASTIE, T., TIBSHIRANI, R., BOTSTEIN, D. and BROWN, P. (2001). Supervised harvesting of expression trees. *Genome Biol.* **2** 0003–1.

ISHWARAN, H., KOGALUR, U., BLACKSTONE, E. and LAUER, M. (2006). Random survival forests. *Ann. Appl. Statist.* **2** 841–860. MR2516796

JOHNS, T., GREGORY, J., INGRAM, W., JOHNSON, C., JONES, A., LOWE, J., MITCHELL, J., ROBERTS, D., SEXTON, D., STEVENSON, D. ET AL. (2003). Anthropogenic climate change for 1860 to 2100 simulated with the HadCM3 model under updated emissions scenarios. *Climate Dynamics* **20** 583–612.

LIAW, A. and WIENER, M. (2002). Classification and regression by random forest. *R News* **2** 18–22.

LIN, Y. and JEON, Y. (2006). Random forests and adaptive nearest neighbors. *J. Amer. Statist. Assoc.* **101** 578–590. MR2256176

MANGASARIAN, O., STREET, W. and WOLBERG, W. (1995). Breast cancer diagnosis and prognosis via linear programming. *Oper. Res.* **43** 570–577. MR1356410

MEINSHAUSEN, N. (2009). Forest Garrote. *Electron. J. Statist.* **3** 1288–1304. MR2566188

NASH, W., SELLERS, T., TALBOT, S., CAWTHORN, A. and FORD, W. (1994). The population biology of abalone in Tasmania. Technical report, Sea Fisheries Division.

OAKLEY, J. and O'HAGAN, A. (2004). Probabilistic sensitivity analysis of complex models: A Bayesian approach. *J. Roy. Statist. Soc. Ser. B* **66** 751–769. MR2088780

R DEVELOPMENT CORE TEAM (2005). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

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.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the Lasso. *J. Roy. Statist. Soc. Ser. B* **58** 267–288. MR1379242

WOLPERT, D. (1992). Stacked generalization. *Neural Networks* **5** 241–259.

YU, B. and BÜHLMANN, P. (2003). Boosting with the L2 loss: Regression and classification. *J. Amer. Statist. Assoc.* **98** 324–339. MR1995709

DEPARTMENT OF STATISTICS
UNIVERSITY OF OXFORD
1 SOUTH PARKS ROAD
OXFORD OX1 3TG
UK
E-MAIL: meinshausen@stats.ox.ac.uk