

Predictive Learning via Rule Ensembles

Jerome H. Friedman

Bogdan E. Popescu

Stanford University

PREDICTIVE LEARNING

y = “response”, “output” variable (unknown)

$\mathbf{x} = (x_1, x_2, \dots, x_n)$ = “input”, “predictor” variables

Prediction: $\hat{y} = F(\mathbf{x})$

$L(y, \hat{y})$ = loss criterion

Lack of accuracy (“risk”):

$$R(F) = E_{\mathbf{x}y}L(y, F(\mathbf{x}))$$

EXAMPLE

$$y \in \{signal, background\} = \{S, B\}$$

$$\mathbf{x} = \{\text{event measured variables}\}$$

$$\hat{y} = F(\mathbf{x}) \in \{S, B\}$$

$$L(y, \hat{y}) = \begin{cases} L_S & \text{if } y = S \ \& \ \hat{y} = B \\ L_B & \text{if } y = B \ \& \ \hat{y} = S \end{cases}$$

$R(F)$ not continuous in model parameters

\Rightarrow search for minimum difficult.

Numeric prediction

$$y = +1 \Rightarrow S, \quad y = -1 \Rightarrow B$$

$$\hat{y} = F(\mathbf{x}) = \text{numerical score:}$$

$$\sim L_S \Pr(y = +1 | \mathbf{x}) - L_B \Pr(y = -1 | \mathbf{x})$$

$$\hat{y} > 0 \Rightarrow S, \quad \hat{y} < 0 \Rightarrow B \quad \text{with confidence} \sim |\hat{y}|$$

$$L(y, \hat{y}) = \text{numerical criterion}$$

$$\text{squared-error: } (y - \hat{y})^2$$

$$\text{logistic regression: } \log(1 + e^{-y \cdot \hat{y}})$$

$$\text{AdaBoost: } e^{-y \cdot \hat{y}}$$

$$\text{RuleFit: } (y - \min(1, \max(-1, \hat{y})))^2$$

Optimal (“target”) function:

$$F^* = \arg \min_F E_{\mathbf{x}y} L(y, F(\mathbf{x}))$$

Unknown, since $p(\mathbf{x}, y)$ unknown

Learning: $T = \{\mathbf{x}_i, y_i, w_i\}_1^N$ “training” sample

\mathbf{x}_i = event measurements

$y_i = \pm 1$ for event $(i) = S/B$

$w_i = L_{S/B} \pi_{S/B} / N_{S/B}$ for event $(i) = S/B$

Goal: approximate $F^*(\mathbf{x})$ by

$F(\mathbf{x}) \leftarrow$ learning procedure (T)

ENSEMBLE LEARNING

$$F(\mathbf{x}) = a_0 + \sum_{m=1}^M a_m f_m(\mathbf{x})$$

$\{f_m(\mathbf{x})\}_1^M =$ basis functions (“base learners”)

Base learner: $f_m(\mathbf{x}) = f(\mathbf{x}; \mathbf{p}_m)$

$\mathbf{p} = (p_1, p_2, \dots) =$ parameters

$\{f(\mathbf{x}; \mathbf{p})\}_{\mathbf{p} \in P} =$ function class

Methods differ: $f(\mathbf{x}; \mathbf{p})$

select: $\{f_m(\mathbf{x})\}_1^M \subset \{f(\mathbf{x}; \mathbf{p})\}_{\mathbf{p} \in P}$

determine: $\{a_m\}_0^M$

GENERIC ENSEMBLE GENERATION PROC. (EGP)

$$F_0(\mathbf{x}) = 0$$

For $m = 1$ to M {

$$\mathbf{p}_m = \arg \min_{\mathbf{p}}$$

$$\sum_{i \in S_m(\eta)} L(y_i, F_{m-1}(\mathbf{x}_i) + f(\mathbf{x}_i; \mathbf{p}))$$

$$f_m(\mathbf{x}) = f(\mathbf{x}; \mathbf{p}_m)$$

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \cdot f_m(\mathbf{x})$$

}

$$\text{ensemble} = \{f_m(\mathbf{x})\}_1^M$$

EGP CONTROL PARAMETERS (FP 2003)

$S_m(\eta)$ = random subsample of size $\eta \leq N$

$\eta \downarrow \Rightarrow$ ensemble diversity \uparrow and comp. \downarrow

Auxiliary “memory” function: step m

$$F_{m-1}(\mathbf{x}) = \nu \cdot \sum_{k=1}^{m-1} f_k(\mathbf{x})$$

retains info $\{f_k(\mathbf{x})\}_1^{m-1}$

$0 \leq \nu \leq 1$ = “memory control” parameter

POPULAR ENSEMBLE METHODS

Bagging: $L(y, \hat{y}) = (y - \hat{y})^2$, $\nu = 0$, $\eta = N/2$

$a_0 = 0$, $\{a_m = 1/M\}_1^M \Rightarrow$ simple average

Random forests: bagging with randomized trees

AdaBoost: $y \in \{-1, 1\}$; $L(y, \hat{y}) = \exp(-y \cdot \hat{y})$

$\nu = 1$ and $\eta = N$, $\hat{y} = \text{sign}(F_M(\mathbf{x}))$

MART (TreeNet): arbitrary y and $L(y, \hat{y})$

Defaults: $\nu = 0.1$, $\eta = N/2$

$\hat{y} = F_M(\mathbf{x})$

ISLE (FP 2003): $F(\mathbf{x}) = \hat{a}_0 + \sum_{m=1}^M \hat{a}_m f_m(\mathbf{x})$

Lasso regression y on $\{f_m(\mathbf{x})\}_1^M$:

$$\{\hat{a}_m\}_0^M = \arg \min_{\{a_m\}_0^M}$$

$$\sum_{i=1}^N L(y_i, a_0 + \sum_{m=1}^M a_m f_m(\mathbf{x}_i))$$

$$+ \lambda \cdot \sum_{m=1}^M |a_m|$$

$\lambda \uparrow \Rightarrow$ more shrinkage and *diversity* of $\{|\hat{a}_m|\}_1^M$

with many $\hat{a}_m = 0$ (selection effect)

estimated by cross-validation

EGP: $(\eta, \nu) = \text{small}$; $\nu \simeq 0.01$, $\eta \sim \sqrt{N}$

Almost all ensemble learning implementations:

Base learners: $f(\mathbf{x}; \mathbf{p}) =$ decision trees

$\mathbf{p} =$ splitting variables and value subsets

defining branches

Reasons:

Desirable data mining properties

Accuracy helped the most

Fast (approximate) algorithms

Here base learners = RULES

Let $x_j \in S_j$ and $s_{jm} \subseteq S_j$

$$f(\mathbf{x}; \mathbf{p}_m) = r_m(\mathbf{x}) = \prod_{j=1}^n I(x_j \in s_{jm}) \in \{0, 1\}$$

Numeric variables: $s_{jm}: t_{jm} < x_j \leq u_{jm}$

Categorical variables: s_{jm} explicitly enumerated

If $s_{jm} = S_j \Rightarrow$ omit x_j factor from rule:

$$r_m(\mathbf{x}) = \prod_{s_{jm} \neq S_j} I(x_j \in s_{jm})$$

$\{x_j \mid s_{jm} \neq S_j\}$ "define" $r_m(\mathbf{x})$

EXAMPLE

$$r_m(\mathbf{x}) = \begin{cases} I(18 \leq \text{age} < 34) \\ \cdot I(\text{marital status} \in \{\text{single, living together} \\ \text{-not married}\}) \\ \cdot I(\text{householder status} = \text{rent}) \end{cases}$$

= 1 \Rightarrow greater odds of visiting bars & night clubs

HEP: rule = intersection of cuts on the variables

RULE GENERATION

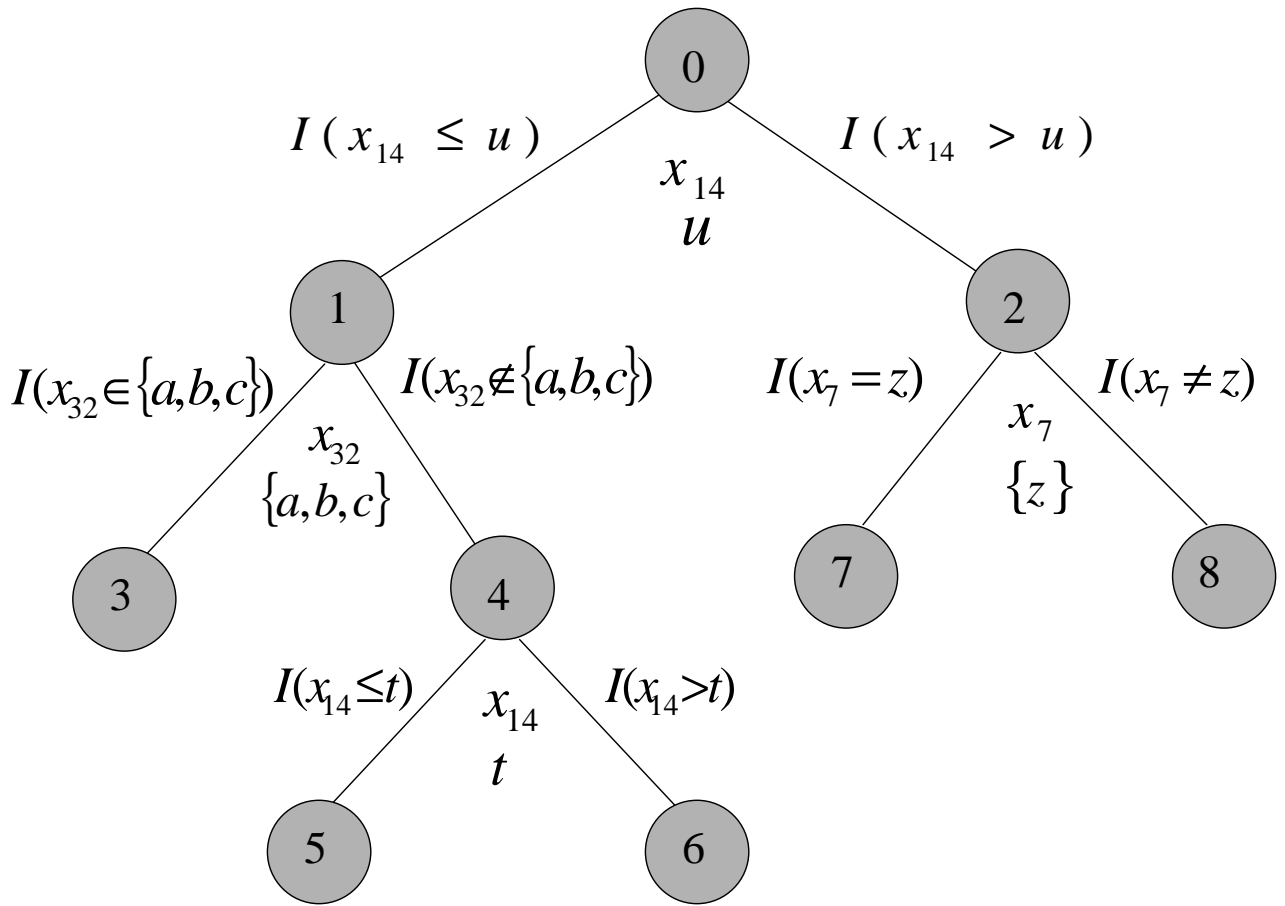
$$f(\mathbf{x}; \mathbf{p}_m) = \prod_{j=1}^n I(x_j \in s_{jm}) \text{ in EGP difficult}$$

Fast algorithms for decision trees \Rightarrow

$$f(\mathbf{x}; \mathbf{p}) = T(\mathbf{x}; \mathbf{p}) = \text{decision tree in EGP}$$

harvest rules from resulting $\{T_m(\mathbf{x})\}_1^M$

All tree nodes (interior and terminal) represent rules



$$r_1(\mathbf{x}) = I(x_{14} \leq u)$$

$$r_6(\mathbf{x}) = I(t < x_{14} \leq u) \cdot I(x_{32} \notin \{a, b, c\})$$

$$r_7(\mathbf{x}) = I(x_{14} > u) \cdot I(x_7 = z).$$

All such rules derived from all trees $\{T_m(\mathbf{x})\}_1^M$

constitute the rule ensemble $\{r_k(\mathbf{x})\}_1^K$

$$K = \sum_{m=1}^M 2(t_m - 1)$$

$t_m = \#$ terminal nodes of T_m

Model: $F(\mathbf{x}) = \hat{a}_0 + \sum_{k=1}^K \hat{a}_k r_k(\mathbf{x})$

$\{\hat{a}_k\}_0^K =$ lasso regression (y on $\{r_k(\mathbf{x})\}_1^K$)

Lasso selection effect \Rightarrow

most ($\sim 80\% - 90\%$) $\hat{a}_k = 0$

ACCURACY

FP 2003: compared tree ensembles

Bag, RF, MART, ISLEs: $\text{EGP}(\nu, \eta) \rightarrow$ lasso

large Monte Carlo + real data

~ 100 data sets, each with different $F^*(\mathbf{x})$

Results: $\text{EGP}(\nu \simeq 0.01, \eta \sim \sqrt{N}) \rightarrow$ lasso best

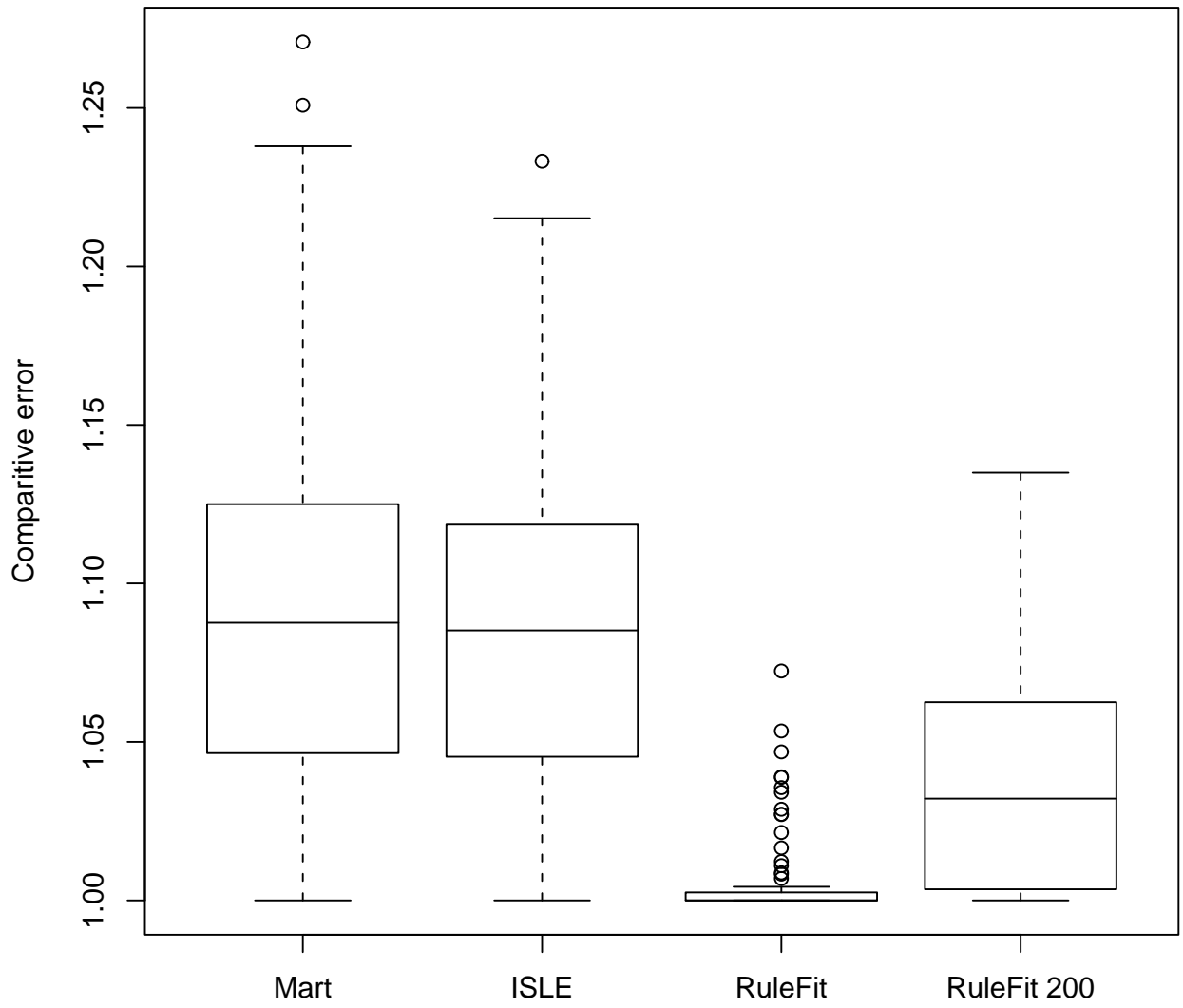
FP (2005): same data sets

RuleFit using same EGP

slightly more accurate $\sim 5\% - 10\%$ than ISLE

Considerably better than others

Classification



TREE SIZE

Controls maximum complexity of rules

$t_m \uparrow \Rightarrow \# \text{ factors in rules } \uparrow$

$\# \text{ factors } \uparrow \Rightarrow \text{ higher order interactions}$

(cross partial derivatives)

$F^*(\mathbf{x}) \sim \text{ high order interactions } \Rightarrow \text{ larger trees}$

$F^*(\mathbf{x}) \sim \text{ main effects and/or low order}$

interaction \Rightarrow smaller trees

Our strategy: tree size $t_m = \text{random}$

$$t_m \sim \exp(-t; \bar{L})$$

$\bar{L} = 2 \Rightarrow$ one factor rules $\Rightarrow F(\mathbf{x}) =$ main effects only

$\bar{L} > 2 \Rightarrow$ distribution of tree sizes $\{t_m\}_1^M$

Exp. dist. $\Rightarrow \sim$ uniform ensemble rule complexity

Lasso chooses among them

RULE BASED INTERPRETATION

$F(\mathbf{x}) = \text{linear model in } \{r_k(\mathbf{x})\}_1^K$

Rules easy to interpret

$K \sim 10^3, \#\{\hat{a}_k \neq 0\} \sim 10^2$

Examine most important rules for interpretation

Linear model \Rightarrow rule importance

$$I_k = |\hat{a}_k| \cdot \sqrt{s_k(1 - s_k)}$$

$s_k = \text{support}$

INPUT VARIABLE IMPORTANCE

Most important variables are those that define
most important rules.

Importance of x_j :

$$J_j = \sum_{x_j \in r_k} I_k / m_k$$

I_k = importance of k th rule (containing x_j)

m_k = # variables defining k th rule

INTERPRETATION

Joint dependence of $\hat{F}(\mathbf{x})$ on relevant $\{x_r\}_1^R$

Plot $\hat{F}(x_{r_1}, \dots, x_{r_R})$ vs. $(x_{r_1}, \dots, x_{r_R})$

Works only if $R \lesssim 2, 3$. What if $R > 2, 3$?

PARTIAL DEPENDENCE FUNCTIONS

\mathbf{x}_s = selected subset of input variables

indexed by $s \subset \{1, 2, \dots, n\}$

$$\mathbf{x} = (\mathbf{x}_s, \mathbf{x}_{\setminus s})$$

Partial dep. on \mathbf{x}_s : $F_s(\mathbf{x}_s) = E_{\mathbf{x}_{\setminus s}}[F(\mathbf{x}_s, \mathbf{x}_{\setminus s})]$

Estimate: $\hat{F}_s(\mathbf{x}_s) = \frac{1}{N} \sum_{i=1}^N F(\mathbf{x}_s, \mathbf{x}_{i\setminus s})$

$\{\mathbf{x}_{i\setminus s}\}_1^N$ = data values of $\mathbf{x}_{\setminus s}$

$F_s(\mathbf{x}_s)$ = “effect” of \mathbf{x}_l on $\hat{F}(\mathbf{x})$ after *accounting*

for (average) effects of $\mathbf{x}_{\setminus l}$

NOT effect of \mathbf{x}_l on $\hat{F}(\mathbf{x})$ *ignoring* $\mathbf{x}_{\setminus l}$

ILLUSTRATION

SIGNAL / BACKGROUND SEPARATION

Training data:

25000 signal events; 25000 background events

Validation (test) data:

11500 signal events; 11500 background events

50 input variables

Parameters (defaults):

$$\nu = 0.01, \quad \eta = \min(N/2, 100 + 6\sqrt{N}) \simeq 0.03$$

Ave. tree size: $\bar{L} = 4$ terminal nodes

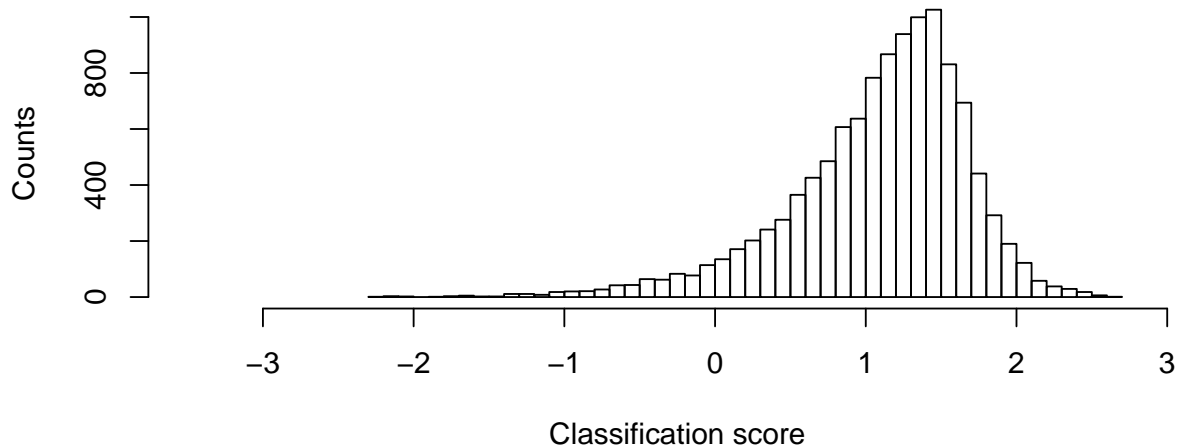
$$K = 3500 \text{ rules} \Rightarrow M = 585 \text{ trees}$$

RuleFit model: 410 rules

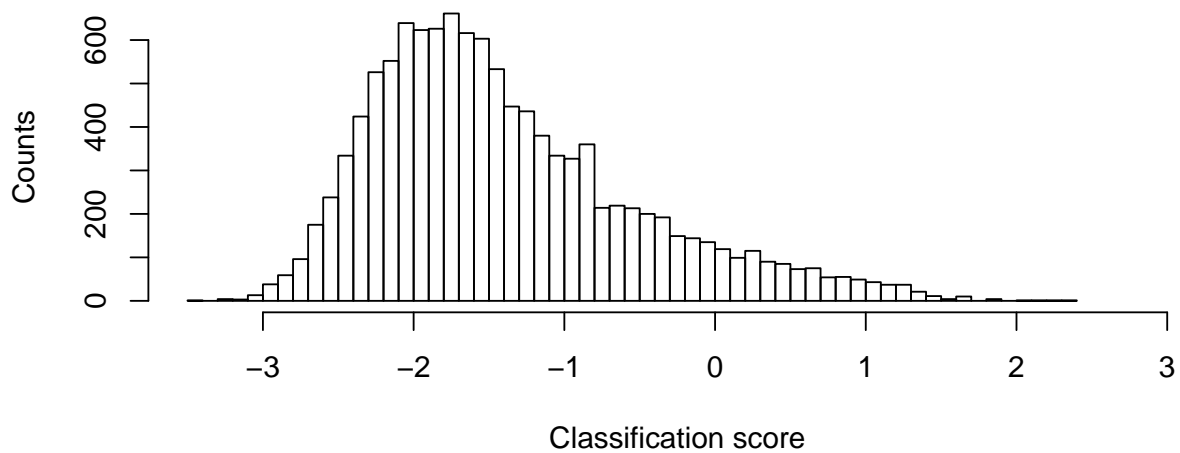
Test set:

$$\text{AUC} = 0.977; \quad \text{error rate} = 6.97\%$$

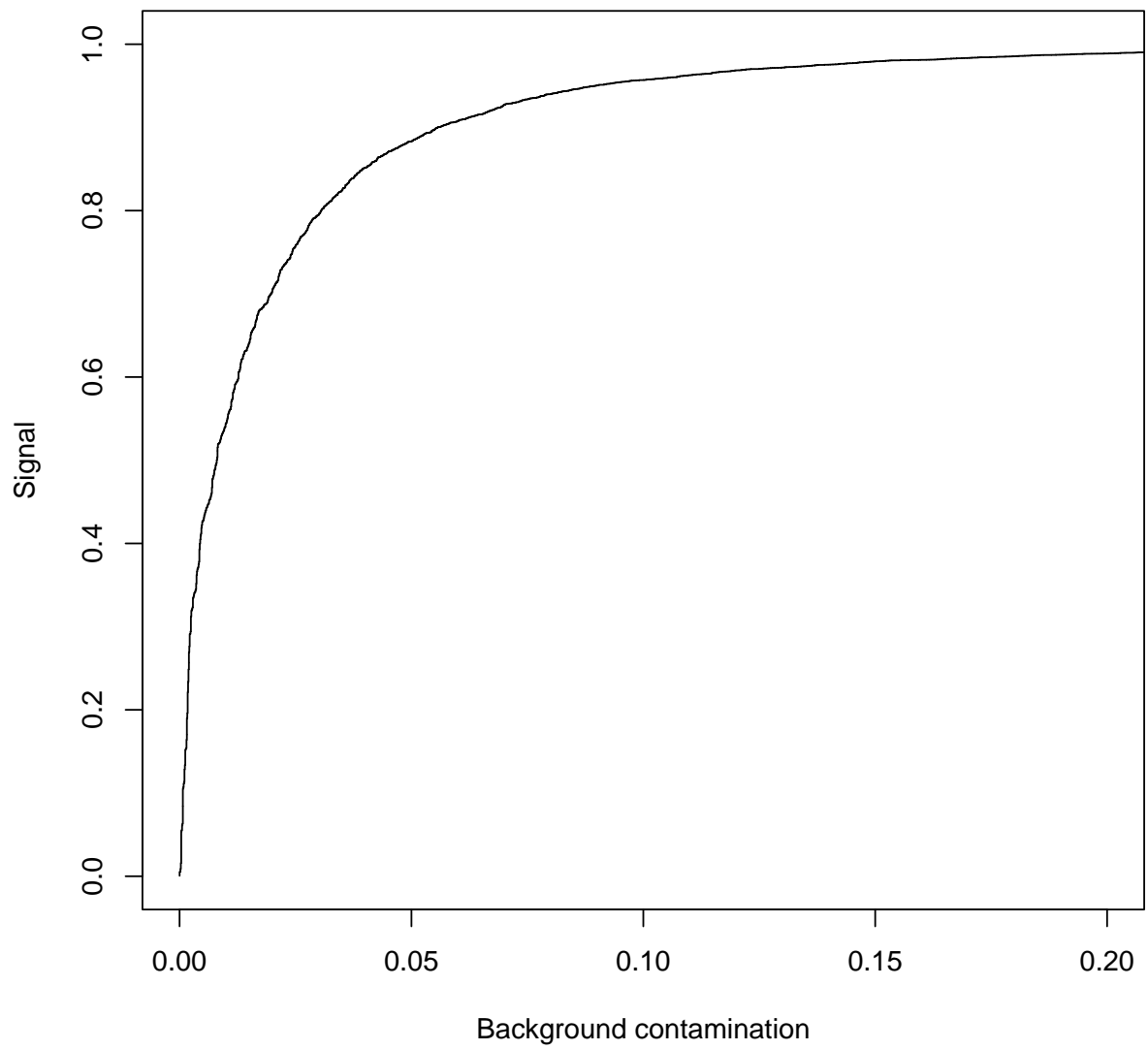
Signal



Background



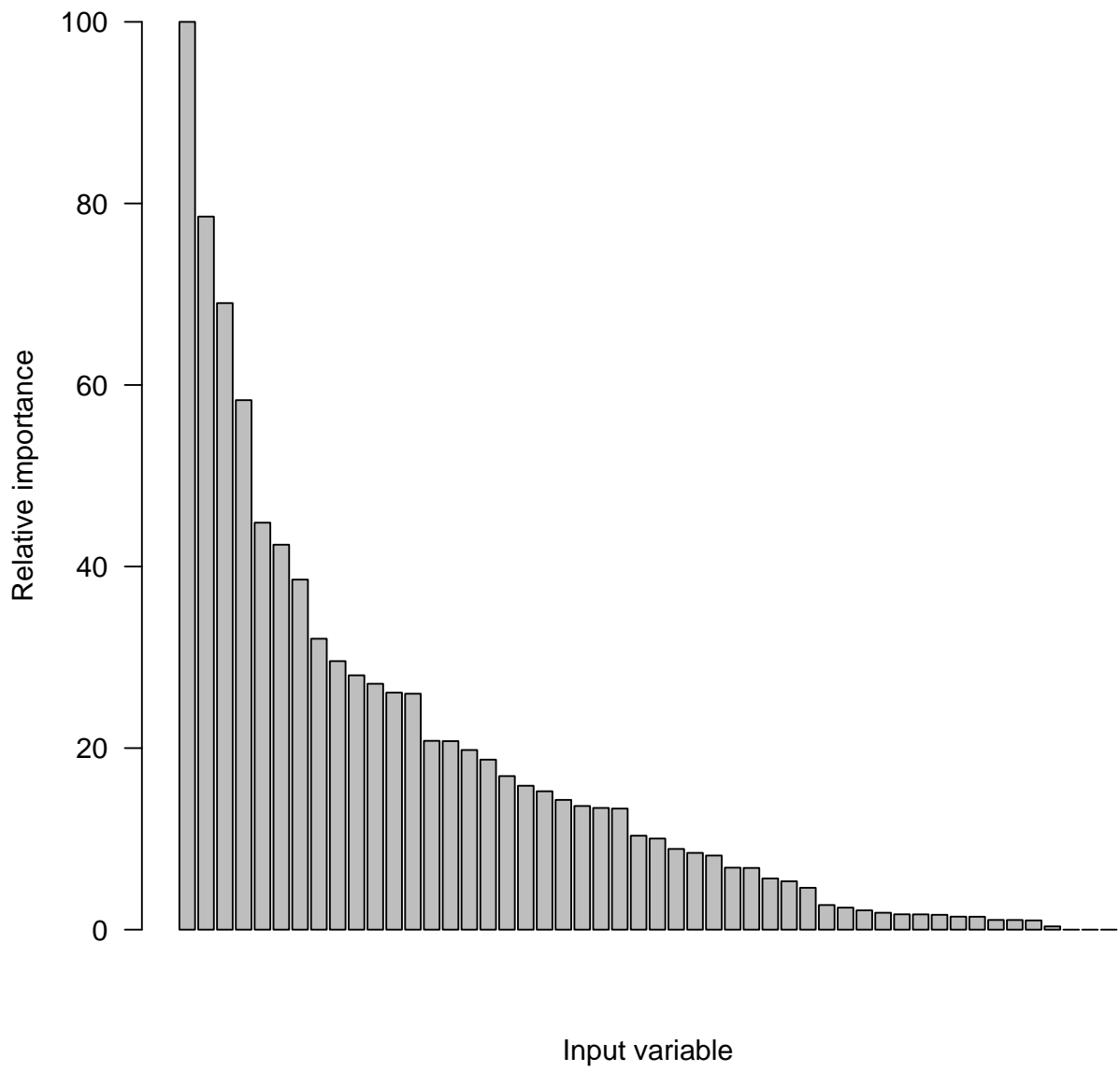
Signal / Background ROC curve



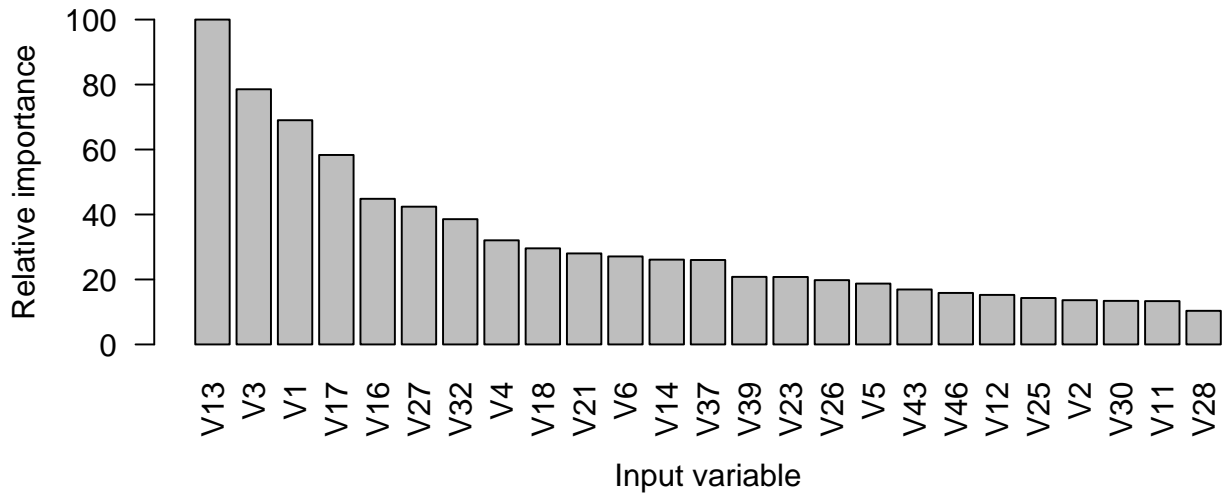
Top five rules

Imp.	Coeff.	sup.	Rule
100	-0.16	0.45	$x_6 \leq 0.31$ & $x_{16} \leq 1117$ & $x_{32} \leq 1.31$
83	0.13	0.41	$0.025 \leq x_{14} < 0.53$ & $x_{27} < 82.4$
82	0.22	0.093	$-500 \leq x_3 < 92.6$ & $x_{21} \leq -0.022$ & $x_{39} > 1.18$
75	0.12	0.32	$x_1 \leq 5.2$ & $-500 \leq x_3 < 92.6$ & $x_{21} > -0.022$
73	-0.12	0.41	$x_1 > 4.37$ & $x_{23} \leq 160.1$ & $x_{32} \leq 1.41$

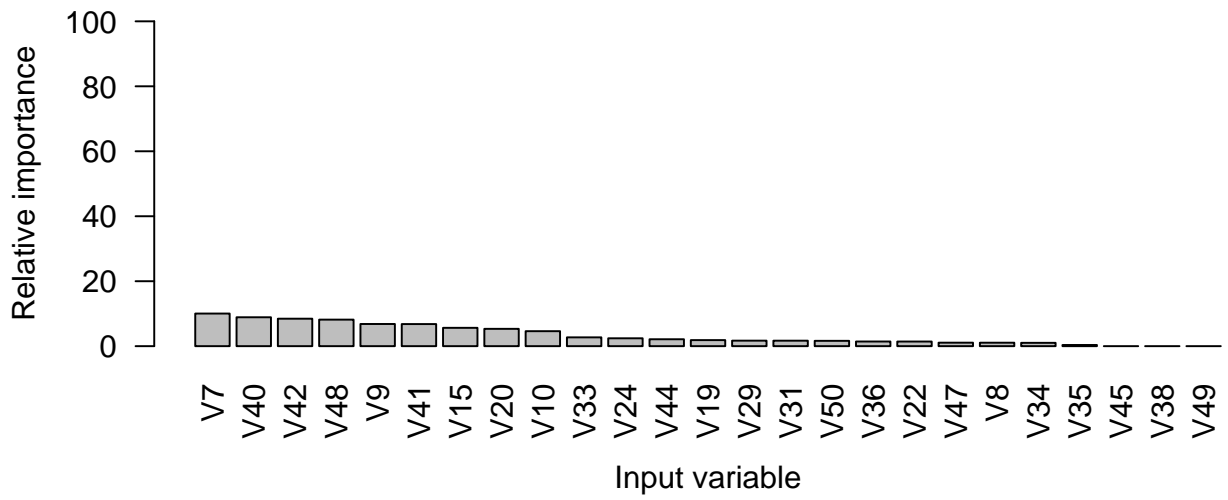
Input variable importances



Top 25 variables



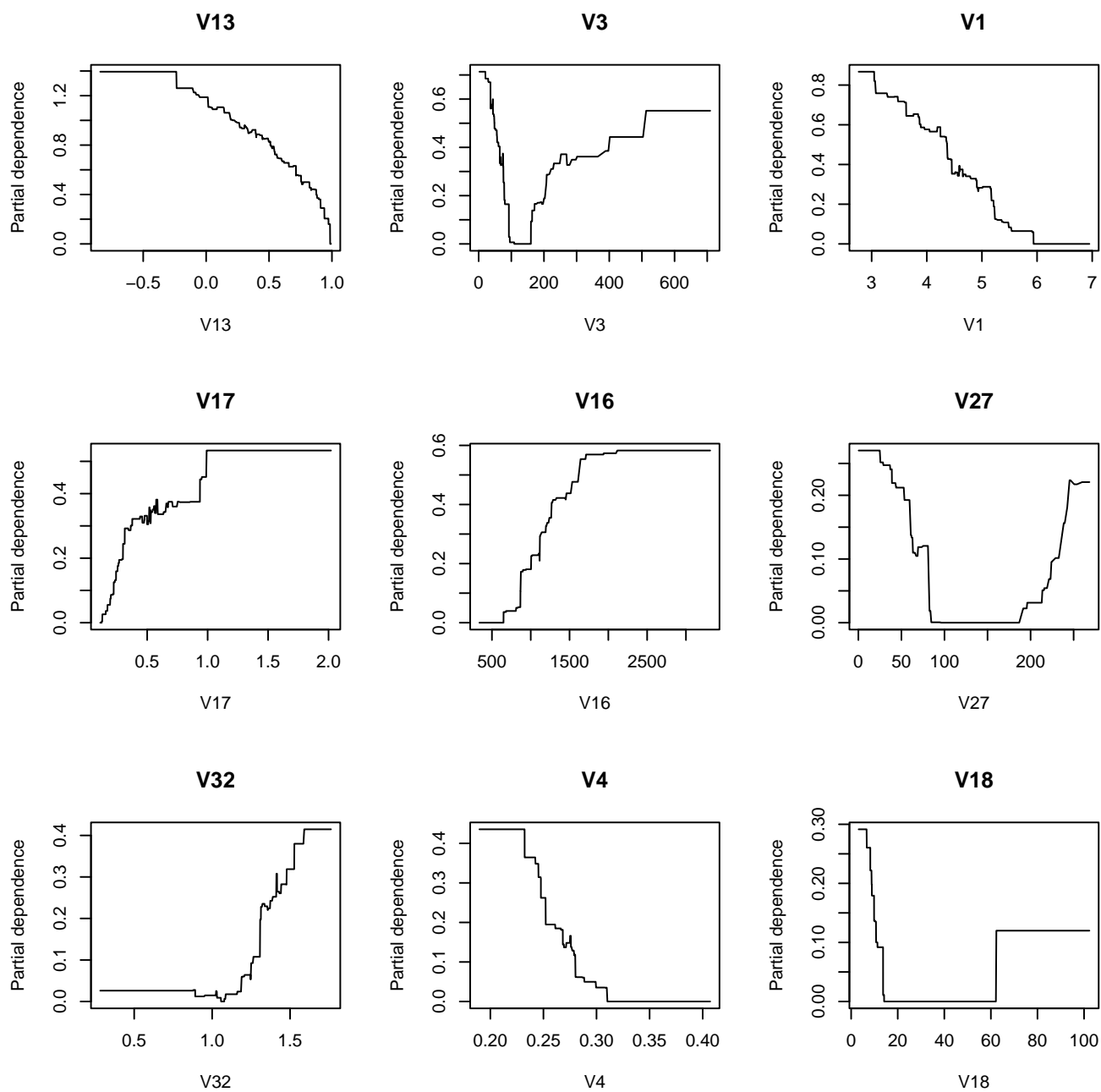
Bottom 25 variables



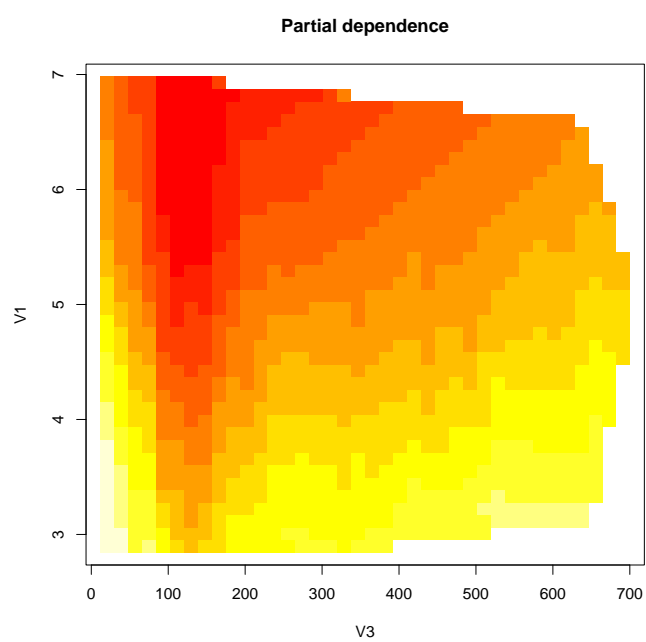
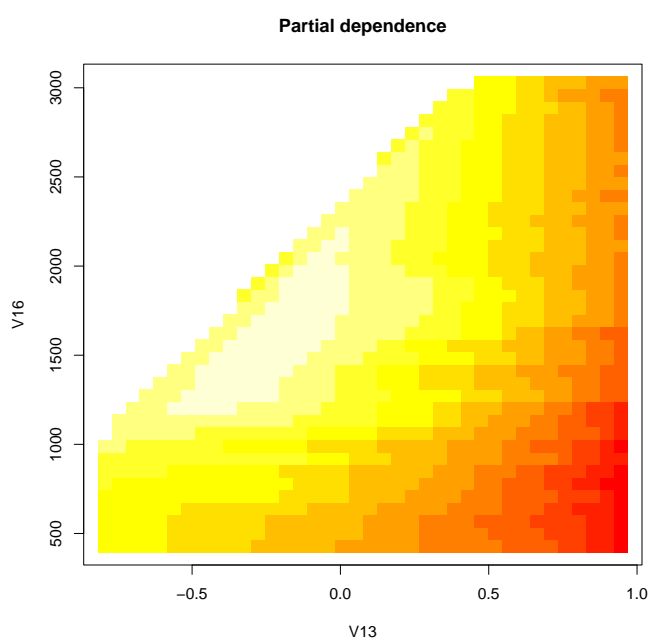
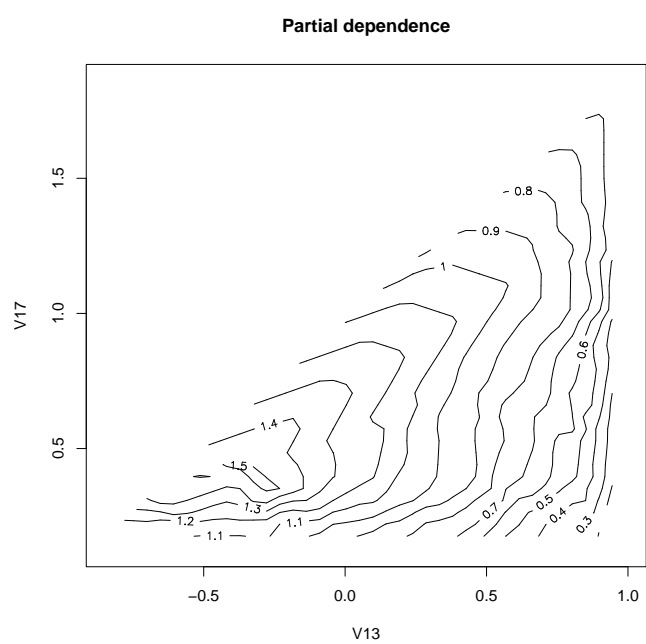
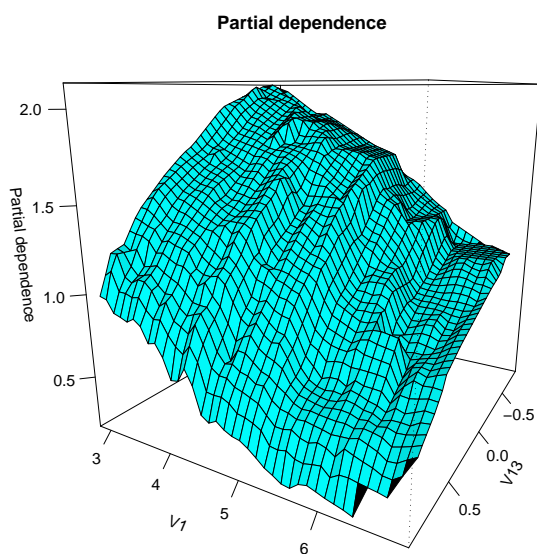
INPUT VARIABLE SELECTION

Vars	1 – AUC	Error
50	0.0230	6.97
25	0.0232	7.06
20	0.0237	7.06
15	0.0264	7.60

Single-variable partial dependences



Two-variable partial dependences



Future Work: rule summarization

Bibliography

Talk:

<http://www-stat.stanford.edu/~jhf/talks/oxford.pdf>

Paper:

<http://www-stat.stanford.edu/~jhf/ftp/RuleFit.pdf>

Software (R interface):

<http://www-stat.stanford.edu/~jhf/RuleFit.html>