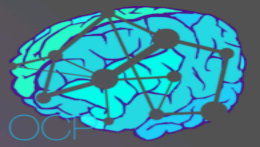
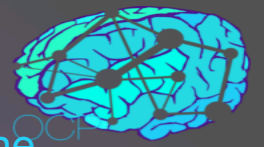


Randomer Forests

Tyler Tomita
Department of Biomedical Engineering
Johns Hopkins University

Motivation



Random forests (RF) has empirically been shown to be the best black-box machine learning method to date

Why does it work so well?

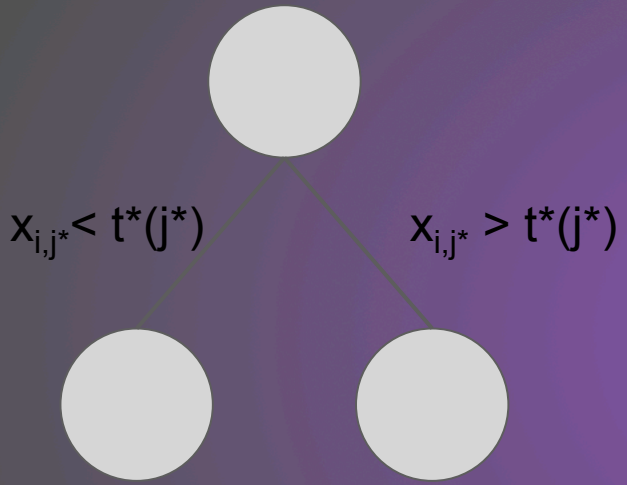
- Real data tends to be sparse (few useful features). Random forest only operates on single features when making recursive splits in decision trees. In other words, recursive splits are sparse.

Can we do better?

- Yes. In addition to being sparse, real data tends to have features that individually are not very informative, but when “mixed” they are.
- Since random forest only finds axis-aligned splits, it may fail in cases where individual features don’t provide much discriminatory information, and an oblique decision forest may be preferred.
- Existing oblique decision forests take dense linear combinations of variables and tend to underperform on sparse data. Furthermore, they are expensive to compute.

Goal: Develop an oblique decision forest that works well on sparse data

Challenge



Generalized scheme for decision trees:

X is an n x p data matrix, where n is the number of observations and p is the number of features.

Y is an n-vector of class labels associated with X

A is a random p x d projection matrix with some distribution f_A

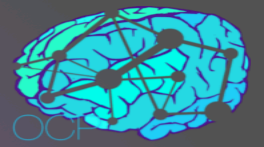
At each node k, sample $A^{(k)}$ and compute $X'^{(k)} = X^{(k)}A^{(k)}$

Find column j* in $X'^{(k)}$ that maximizes class purity in child nodes and corresponding threshold value t*(j*)

The distribution f_A dictates the possible split directions in Cartesian space. RF constrains A to having a single nonzero in each column. This constraint only allows for axis-aligned splits.

Challenge: Finding an f_A that allows oblique splits, empirically performs well, and maintains the space and time complexity of RF

Solution



Sample elements of A i.i.d. from a sparse Rademacher distribution:

$p(a_{i,j} = 1) = p(a_{i,j} = -1) = 1/(2s)$

$p(a_{i,j} = 0) = 1 - 1/s$

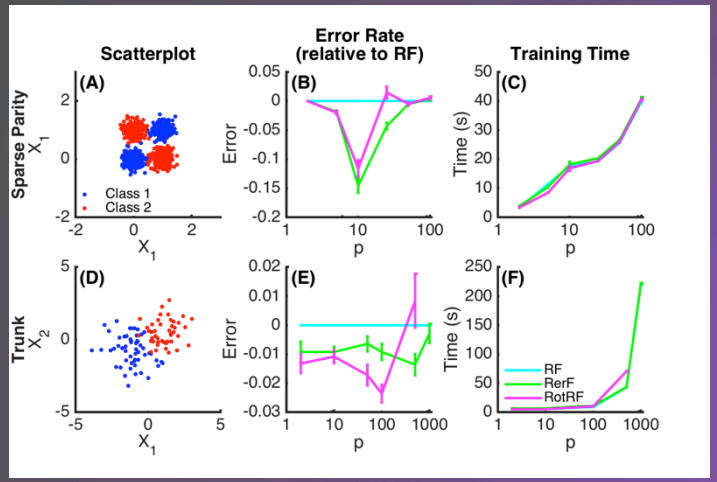
$A_{RF} =$

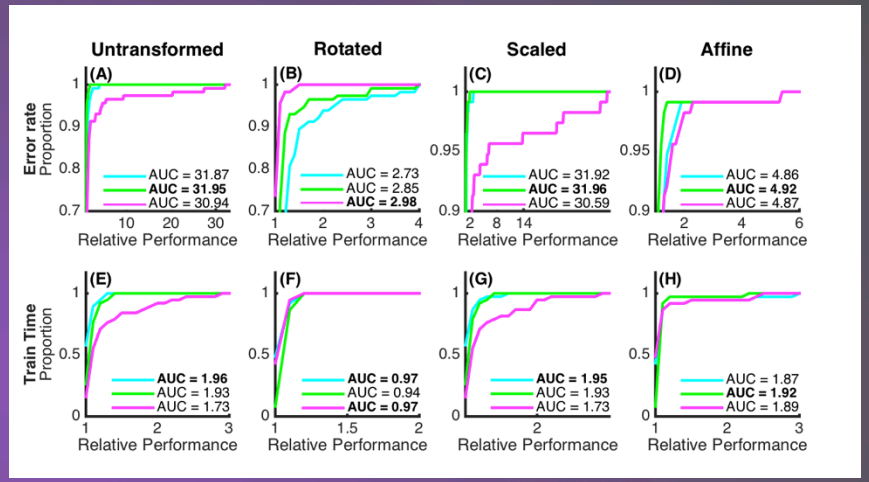
1	0	0
0	1	0
0	0	0
0	0	1
0	0	0

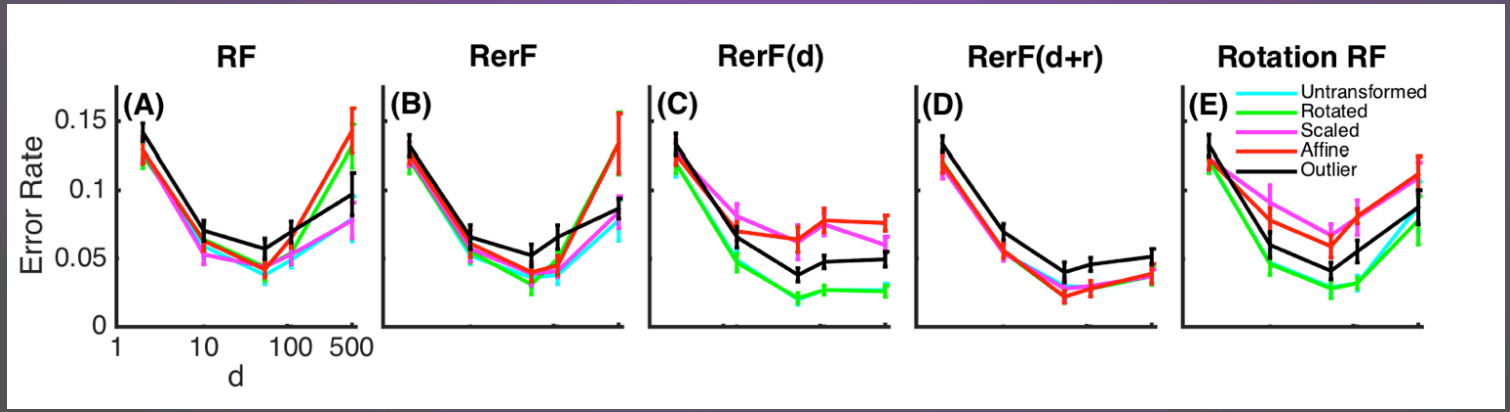
$A_{RerF} =$

0	1	0
1	0	0
-1	1	0
0	0	1
0	0	0

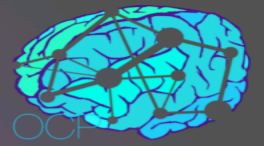
Results


Scatterplot
Error Rate (relative to RF)
Training Time
Sparse Parity
Trunk
(A)
(B)
(C)
(D)
(E)
(F)
Class 1
Class 2
Error
Time (s)
p
RF
RerF
RotRF


Untransformed
Rotated
Scaled
Affine
Error rate
Proportion
Train Time
Proportion
Relative Performance
AUC = 31.87
AUC = 31.95
AUC = 30.94
AUC = 2.73
AUC = 2.85
AUC = 2.98
AUC = 31.92
AUC = 31.96
AUC = 30.59
AUC = 4.86
AUC = 4.92
AUC = 4.87
AUC = 1.96
AUC = 1.93
AUC = 1.73
AUC = 0.97
AUC = 0.94
AUC = 0.97
AUC = 1.95
AUC = 1.93
AUC = 1.73
AUC = 1.87
AUC = 1.92
AUC = 1.89


RF
RerF
RerF(d)
RerF(d+r)
Rotation RF
Error Rate
d
Untransformed
Rotated
Scaled
Affine
Outlier

Conclusions and future directions



In Summary:

We have proposed a novel decision forest method which we call randomer forest (RerF)

We have demonstrated that RerFs are especially well-suited for classification problems in which axis-aligned splits are suboptimal, and at the same time, have a large number of irrelevant features relative to relevant ones.

RerFs empirically outperform RF and another existing oblique method on a suite of >100 benchmark datasets

Future directions:

Further tune the distribution f_A to improve empirical performance

Develop a more efficient implementation of RerF using FlashR

Extend consistency theory for RF to RerF

Extend the application of RerF to other exploitation tasks such as regression, density estimation, etc.