Random Forests for Regression and Classification

Outline

- Background.
- Trees.
- Bagging.
- Random Forests.
- Variable importance.
- Partial dependence plots and interpretation of effects.
- Proximity.
- Visualization.
- New developments.

Two Natural Questions

1. ***Why bootstrap? (Why subsample?)***

   Bootstrapping → out-of-bag data →
   - Estimated error rate and confusion matrix
   - Variable importance

2. ***Why trees?***

   Trees → proximities →
   - Missing value fill-in
   - Outlier detection
   - Illuminating pictures of the data (clusters, structure, outliers)

The RF Predictor

- A case in the training data is *not* in the bootstrap sample for about one third of the trees (we say the case is “out of bag” or “oob”). Vote (or average) the predictions of *these trees* to give ***the RF predictor.***

- The ***oob error rate*** is the error rate of the ***RF predictor.***

- The ***oob confusion matrix*** is obtained from the ***RF predictor.***

- For new cases, vote (or average) *all* the trees to get the ***RF predictor.***

The RF Predictor

For example, suppose we fit 1000 trees, and a case is out-of-bag in 339 of them, of which:

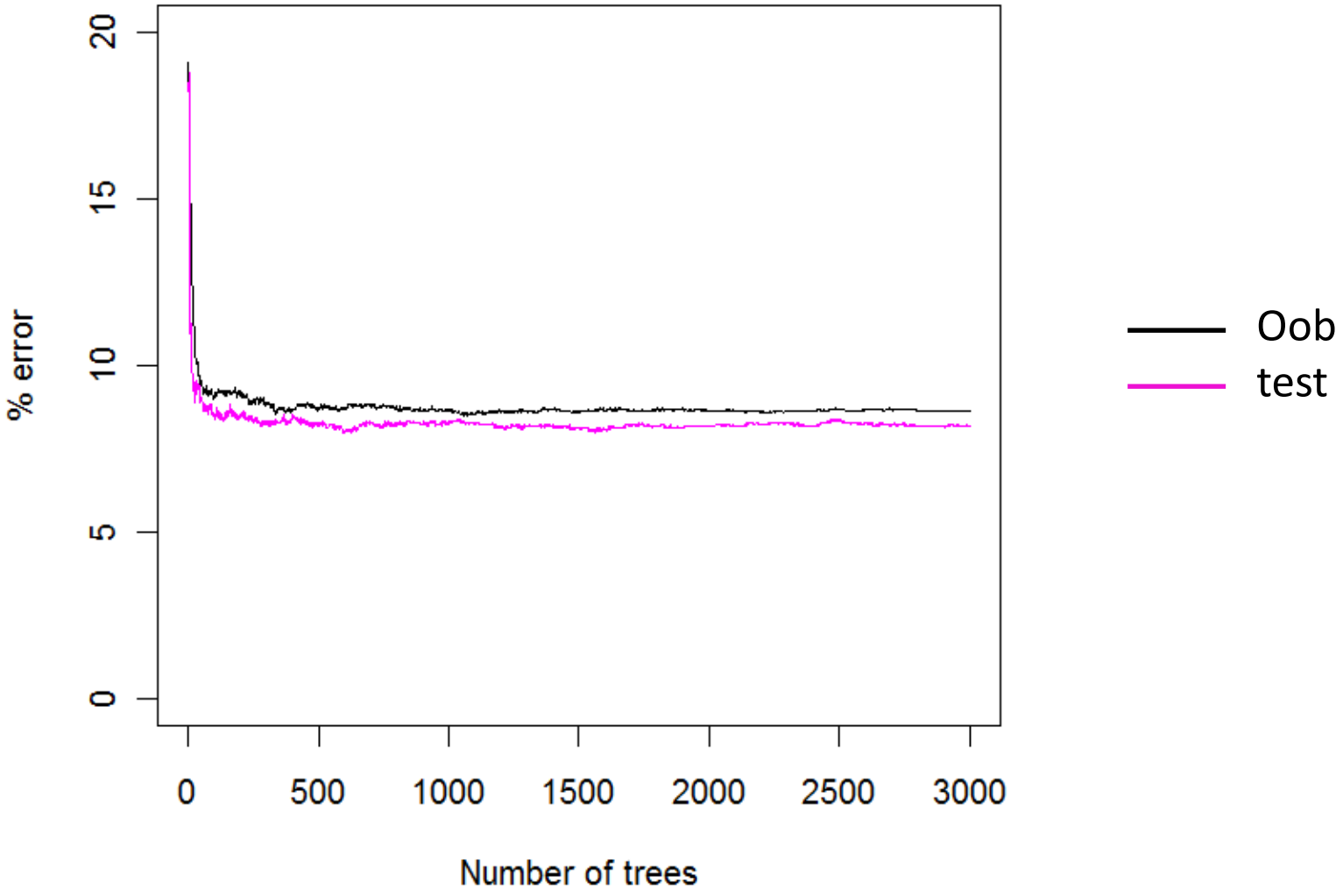
283 say “class 1”

56 say “class 2”

The RF predictor for this case is class 1.

The “oob” error gives an estimate of test set error (generalization error) as trees are added to the ensemble.

RFs do not overfit as we fit more trees



RF handles thousands of predictors

Ramón Díaz-Uriarte, Sara Alvarez de Andrés
Bioinformatics Unit, Spanish National Cancer Center
March, 2005 http://ligarto.org/rdiaz

Compared

- SVM, linear kernel
- KNN/crossvalidation (Dudoit et al. JASA 2002)
- DLDA
- Shrunken Centroids (Tibshirani et al. PNAS 2002)
- Random forests

"Given its performance, random forest and variable selection using random forest should probably become part of the standard tool-box of methods for the analysis of microarray data."

Microarray Datasets

Data	*P*	*N*	*# Classes*
Leukemia	3051	38	2
Breast 2	4869	78	2
Breast 3	4869	96	3
NCI60	5244	61	8
Adenocar	9868	76	2
Brain	5597	42	5
Colon	2000	62	2
Lymphoma	4026	62	3
Prostate	6033	102	2
Srbct	2308	63	4

Microarray Error Rates

Data	*SVM*	*KNN*	*DLDA*	*SC*	*RF*	rank
Leukemia	.014	.029	.020	.025	.051	**5**
Breast 2	.325	.337	.331	.324	.342	**5**
Breast 3	.380	.449	.370	.396	.351	**1**
NCI60	.256	.317	.286	.256	.252	**1**
Adenocar	.203	.174	.194	.177	.125	**1**
Brain	.138	.174	.183	.163	.154	**2**
Colon	.147	.152	.137	.123	.127	**2**
Lymphoma	.010	.008	.021	.028	.009	**2**
Prostate	.064	.100	.149	.088	.077	**2**
Srbct	.017	.023	.011	.012	.021	**4**
Mean	**.155**	**.176**	**.170**	**.159**	**.151**	

RF handles thousands of predictors

- Add noise to some standard datasets and see how well Random Forests:
  - predicts
  - detects the important variables

RF error rates (%)

	No noise added	*10 noise variables*		*100 noise variables*	
Dataset	***Error rate***	***Error rate***	***Ratio***	***Error rate***	***Ratio***
breast	3.1	2.9	**0.93**	2.8	**0.91**
diabetes	23.5	23.8	**1.01**	25.8	**1.10**
ecoli	11.8	13.5	**1.14**	21.2	**1.80**
german	23.5	25.3	**1.07**	28.8	**1.22**
glass	20.4	25.9	**1.27**	37.0	**1.81**
image	1.9	2.1	**1.14**	4.1	**2.22**
iono	6.6	6.5	**0.99**	7.1	**1.07**
liver	25.7	31.0	**1.21**	40.8	**1.59**
sonar	15.2	17.1	**1.12**	21.3	**1.40**
soy	5.3	5.5	**1.06**	7.0	**1.33**
vehicle	25.5	25.0	**0.98**	28.7	**1.12**
votes	4.1	4.6	**1.12**	5.4	**1.33**
vowel	2.6	4.2	**1.59**	17.9	**6.77**

RF error rates

Error rates (%)		*Number of noise variables*			
Dataset	*No noise added*	*10*	*100*	*1,000*	*10,000*
breast	3.1	2.9	2.8	3.6	8.9
glass	20.4	25.9	37.0	51.4	61.7
votes	4.1	4.6	5.4	7.8	17.7

Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.
- Partial plots and interpretation of effects.

Variable Importance

RF computes two measures of variable importance, one based on a rough-and-ready measure (Gini for classification) and the other based on permutations.

To understand how permutation importance is computed, need to understand local variable importance. But first…

RF variable importance

		10 noise variables		*100 noise variables*	
Dataset	***m***	***Number in top m***	***Percent***	***Number in top m***	***Percent***
breast	9	9.0	**100.0**	9.0	**100.0**
diabetes	8	7.6	**95.0**	7.3	**91.2**
ecoli	7	6.0	**85.7**	6.0	**85.7**
german	24	20.0	**83.3**	10.1	**42.1**
glass	9	8.7	**96.7**	8.1	**90.0**
image	19	18.0	**94.7**	18.0	**94.7**
ionosphere	34	33.0	**97.1**	33.0	**97.1**
liver	6	5.6	**93.3**	3.1	**51.7**
sonar	60	57.5	**95.8**	48.0	**80.0**
soy	35	35.0	**100.0**	35.0	**100.0**
vehicle	18	18.0	**100.0**	18.0	**100.0**
votes	16	14.3	**89.4**	13.7	**85.6**
vowel	10	10.0	**100.0**	10.0	**100.0**

RF error rates

Number in top m		*Number of noise variables*			
Dataset	*m*	*10*	*100*	*1,000*	*10,000*
breast	9	9.0	9.0	9	9
glass	9	8.7	8.1	7	6
votes	16	14.3	13.7	13	13

Local Variable Importance

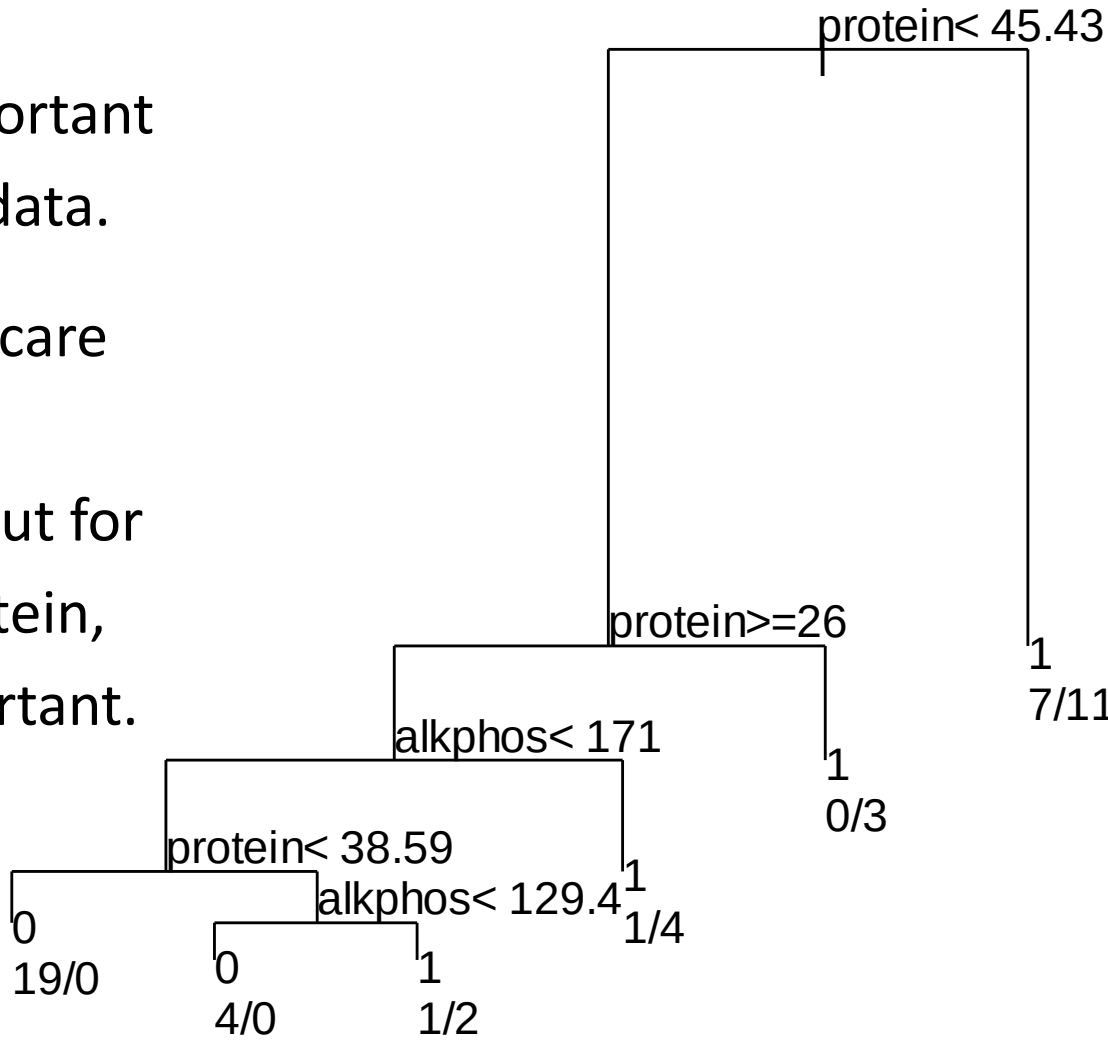
We usually think about variable importance as an overall measure. In part, this is probably because we fit models with global structure (linear regression, logistic regression).

In CART, variable importance is local.

Local Variable Importance

Different variables are important in different regions of the data.

If protein is high, we don't care about alkaline phosphate. Similarly if protein is low. But for intermediate values of protein, alkaline phosphate is important.



Local Variable Importance

For each tree, look at the out-of-bag data:

- randomly permute the values of variable *j*
- pass these perturbed data down the tree, save the classes.

For case *i* and variable *j* find

{ error rate with variable *j* permuted } – { error rate with no permutation }

where the error rates are taken over all trees for which case *i* is out-of-bag.

Local importance for a class 2 case

TREE	No permutation	Permute variable 1	…	Permute variable m
1	2	2	…	1
3	2	2	…	2
4	1	1	…	1
9	2	2	…	1
…	…	…	…	…
992	2	2	…	2
% Error	10%	11%	…	35%

Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.
- Partial plots and interpretation of effects.

Proximities

Proximity of two cases is the proportion of the time that they end up in the same node.

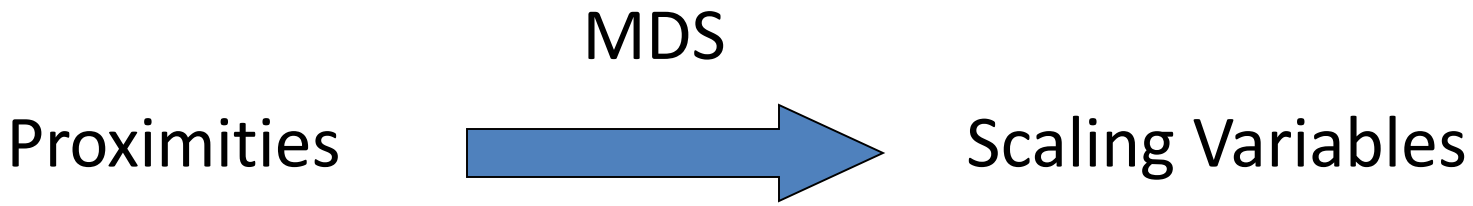
The proximities don't just measure similarity of the variables - they also take into account the importance of the variables.

Two cases that have quite **different** predictor variables might have **large** proximity if they differ only on variables that are **not important**.

Two cases that have quite **similar** values of the predictor variables might have **small** proximity if they differ on inputs that are **important.**

Visualizing using Proximities

To “look” at the data we use classical multidimensional scaling (MDS) to get a picture in 2-D or 3-D:



Might see clusters, outliers, unusual structure.

Can also use nonmetric MDS.

Visualizing using Proximities

- at-a-glance information about which classes are overlapping, which classes differ
- find clusters within classes
- find easy/hard/unusual cases

With a good tool we can also

- identify characteristics of unusual points
- see which variables are locally important
- see how clusters or unusual points differ

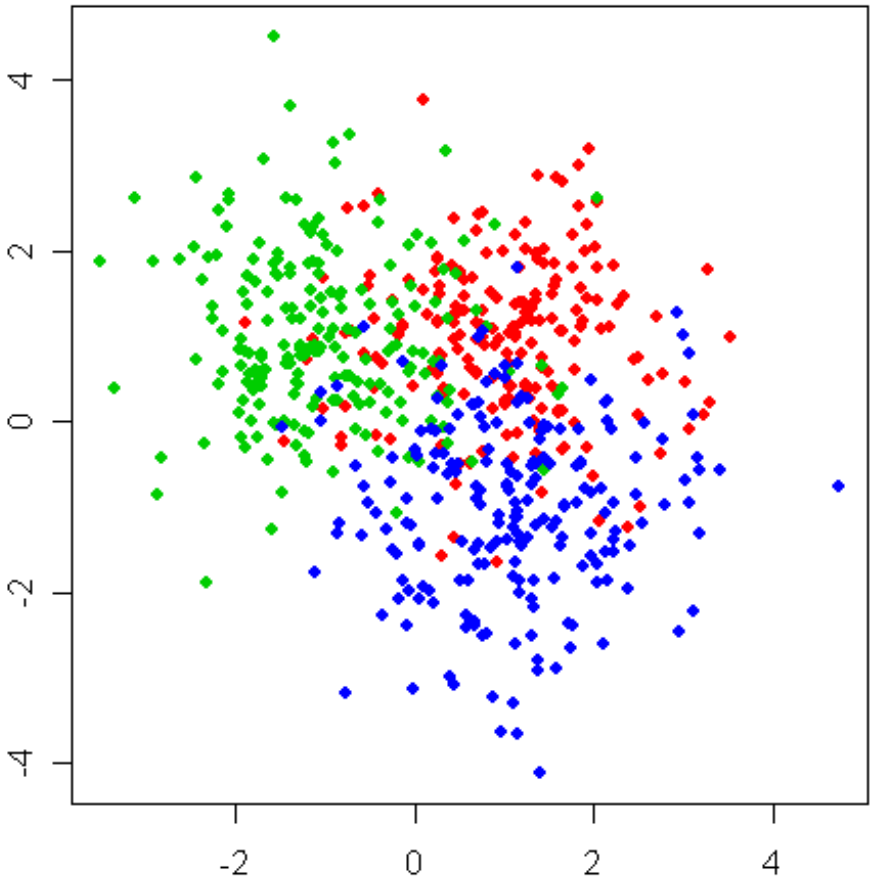
Visualizing using Proximities

Synthetic data, 600 cases

2 meaningful variables

48 “noise” variables

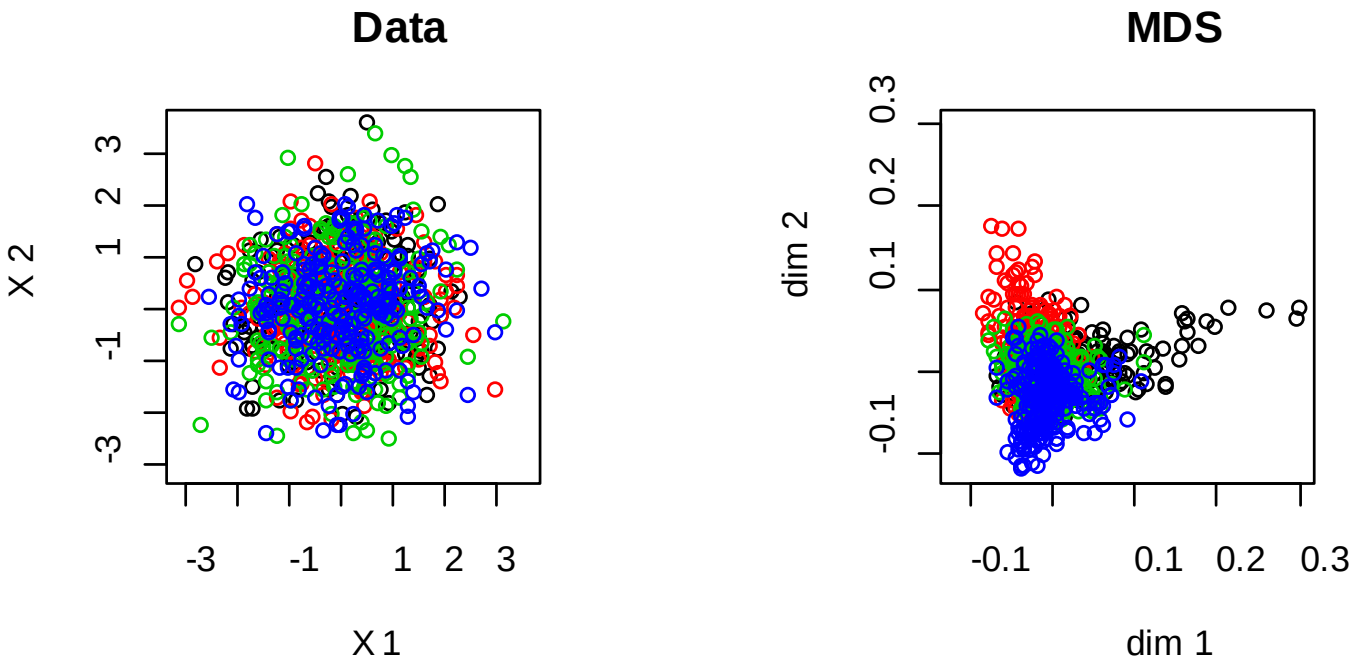
3 classes



The Problem with Proximities

Proximities based on *all* the data overfit!

e.g. two cases from different classes must have proximity zero if trees are grown deep.



Proximity-weighted Nearest Neighbors

RF is like a nearest-neighbor classifier:

- Use the proximities as weights for nearest-neighbors.
- Classify the training data.
- Compute the error rate.

Want the error rate to be close to the RF oob error rate.

BAD NEWS! If we compute proximities from trees in which both cases are OOB, we don't get good accuracy when we use the proximities for prediction!

Proximity-weighted Nearest Neighbors

Dataset	*RF*	*OOB*
breast	2.6	2.9
diabetes	24.2	23.7
ecoli	11.6	12.5
german	23.6	24.1
glass	20.6	**23.8**
image	1.9	2.1
iono	6.8	6.8
liver	26.4	26.7
sonar	13.9	**21.6**
soy	5.1	5.4
vehicle	24.8	**27.4**
votes	3.9	3.7
vowel	2.6	**4.5**

Proximity-weighted Nearest Neighbors

Dataset	*RF*	*OOB*
Waveform	15.5	16.1
Twonorm	3.7	4.6
Threenorm	14.5	15.7
Ringnorm	5.6	5.9

New Proximity Method

Start with P = I, the identity matrix.

For each observation *i*:

For each tree in which case *i* is oob:

- Pass case i down the tree and note which terminal node it falls into.
- Increase the proximity between observation *i* and the *k* in-bag cases that are in the same terminal node, by the amount *1/k*.

Can show that except for ties, this gives the same error rate as RF, when used as a proximity-weighted nn classifier.

New Method

Dataset	*RF*	*OOB*	*New*
breast	2.6	2.9	2.6
diabetes	24.2	23.7	24.4
ecoli	11.6	12.5	11.9
german	23.6	24.1	23.4
glass	20.6	**23.8**	20.6
image	1.9	2.1	1.9
iono	6.8	6.8	6.8
liver	26.4	26.7	26.4
sonar	13.9	**21.6**	13.9
soy	5.1	5.4	5.3
vehicle	24.8	**27.4**	24.8
votes	3.9	3.7	3.7
vowel	2.6	**4.5**	2.6

New Method

Dataset	*RF*	*OOB*	*New*
Waveform	15.5	16.1	15.5
Twonorm	3.7	4.6	3.7
Threenorm	14.5	15.7	14.5
Ringnorm	5.6	5.9	5.6

But…

The new “proximity” matrix is not symmetric!

$\rightarrow$ Methods for doing multidimensional scaling on asymmetric matrices.

Other Uses for Random Forests

- Missing data imputation.
- Feature selection (before using a method that cannot handle high dimensionality).
- Unsupervised learning (cluster analysis).
- Survival analysis without making the proportional hazards assumption.

Missing Data Imputation

Fast way: replace missing values for a given variable using the median of the non-missing values (or the most frequent, if categorical)

Better way (using proximities):

1. Start with the fast way.
2. Get proximities.
3. Replace missing values in case **i** by a weighted average of non-missing values, with weights proportional to the proximity between case **i** and the cases with the non-missing values.

Repeat steps 2 and 3 a few times (5 or 6).

Feature Selection

- Ramón Díaz-Uriarte:

  varSelRF R package.

- In the NIPS competition 2003, several of the top entries used RF for feature selection.

Unsupervised Learning

Global histone modification patterns predict risk of prostate cancer recurrence

David B. Seligson, Steve Horvath, Tao Shi, Hong Yu, Sheila Tze, Michael Grunstein and Siavash K. Kurdistan (all at UCLA).

Used RF clustering of 183 tissue microarrays to find two disease subgroups with distinct risks of tumor recurrence.

http://www.nature.com/nature/journal/v435/n7046/full/nature03672.html

Outline

- Background.
- Trees.
- Bagging predictors.
- Random Forests algorithm.
- Variable importance.
- Proximity measures.
- Visualization.

Case Study: Cavity Nesting birds in the Uintah Mountains, Utah

- Red-naped sapsucker (*Sphyrapicus nuchalis*) (*n* = 42 nest sites)

- Mountain chickadee (*Parus gambeli*) (*n* = 42 nest sites)

- Northern flicker (*Colaptes auratus*) (*n* = 23 nest sites)

- *n* = 106 non-nest sites

Case Study: Cavity Nesting birds in the Uintah Mountains, Utah

- Response variable is the presence (coded 1) or absence (coded 0) of a nest.
- Predictor variables (measured on 0.04 ha plots around the sites) are:
  - Numbers of trees in various size classes from less than 1 inch in diameter at breast height to greater than 15 inches in diameter.
  - Number of snags and number of downed snags.
  - Percent shrub cover.
  - Number of conifers.
  - Stand Type, coded as 0 for pure aspen and 1 for mixed aspen and conifer.

Autism

Data courtesy of J.D.Odell and R. Torres, USU

154 subjects (308 chromosomes)

7 variables, all categorical (up to 30 categories)

2 classes:

- **Normal, blue (69 subjects)**
- **Autistic, red (85 subjects)**

Brain Cancer Microarrays

Pomeroy et al. Nature, 2002.
Dettling and Bühlmann, Genome Biology, 2002.

42 cases, 5,597 genes, 5 tumor types:

- **10 medulloblastomas BLUE**
- **10 malignant gliomas PALE BLUE**
- **10 atypical teratoid/rhabdoid tumors (AT/RTs) GREEN**
- **4 human cerebella ORANGE**
- **8 PNETs RED**

Random Forests Software

- Free, open-source code (fortran)

  www.stat.berkeley.edu/~breiman/forests

- R interface, independent development (Andy Liaw and Matthew Wiener)