MONASH University

ETC3250 Business Analytics: Advanced Classification - Trees & Forests

Souhaib Ben Taieb, Di Cook, Rob Hyndman

October 5, 2015

Random forests - overview

- Multiple trees, fit to samples
- Sample cases, using bootstrapping (ones not chosen are called out-of-bag, used for testing purposes)
- Sample variables
- Lots of control parameters
- Lots of diagnostics generated!

Bagging

- Bagging stands for "bootstrap aggregation". Combine the results from multiple models built on different bootstrap samples.
- Random forests are an example of bagging
- Bagging can be used with almost any classifier
- Bagging reduces variation in estimates

Forest algorithm

1. Input: $L = (x_i, y_i), i = 1, ..., n, y_i \in \{1, ..., g\}, m < p$, number of variables chosen for each tree, B is the number of bootstrap samples.
2. For $b = 1, 2, ..., B$:

   - Draw a bootstrap sample, L^{*b} of size n^{*b} from L.
   - Grow tree classifier, T^{*b}. At each node use a random selection of m variables, and grow to maximum depth without pruning.
   - Predict the class of each case not drawn in L^{*b}.

3. Combine the predictions for each case, by majority vote, to give predicted class.

Input defaults

- B is at least 1000
- $m = \sqrt{(p)}$
- n^{*b} is usually about $\frac{2}{3}n$

Error

Compute the proportion of times the case is misclassified when it is out-of-bag (oob). Average these to give the predictive error.

Diagnostics

- Variable importance: more complicated than one might think
- Vote matrix, $n \times g$: Proportion of times a case is predicted to the class k.
- Proximities, $n \times n$: Closeness of cases measured by how often they are in the same terminal node.

Variable importance

1. For every tree predict the oob cases and count the number of votes cast for the correct class.
2. Randomly permute the values on a variable in the oob cases and predict the class for these cases.
3. Subtract the number of votes for the correct class in the variable-permuted oob cases from the number of votes for the correct class in the real oob cases. The average of this number over all trees in the forest is the raw importance score for that variable. If the value is small, then the variable is not very important.

Gini importance

- Gini importance adds up the difference in impurity value of the descendant nodes with the parent node.
- Quick to calculate, and usually consistent with the results of the permutation method.

Vote matrix

- Proportion of trees the case is predicted to be each class, ranges between 0-1
- Can be used to identify troublesome cases.
- Used with plots of the actual data can help determine if it is the record itself that is the problem, or if it is a limitation of the method.
- Understand the difference in accuracy of prediction for different classes.

Proximities

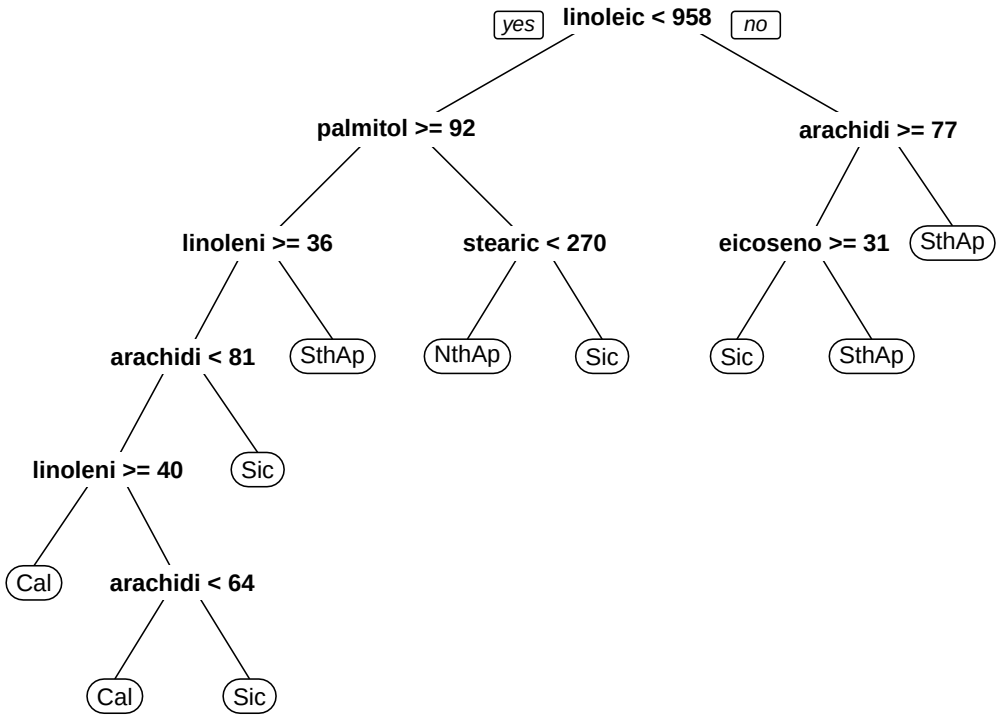
- Run both in- and out-of-bag cases down the tree, and increase proximity value of cases i, j by 1 each time they are in the same terminal node.
- Normalize by dividing by B.

Example: Fit tree to olive samples from the south

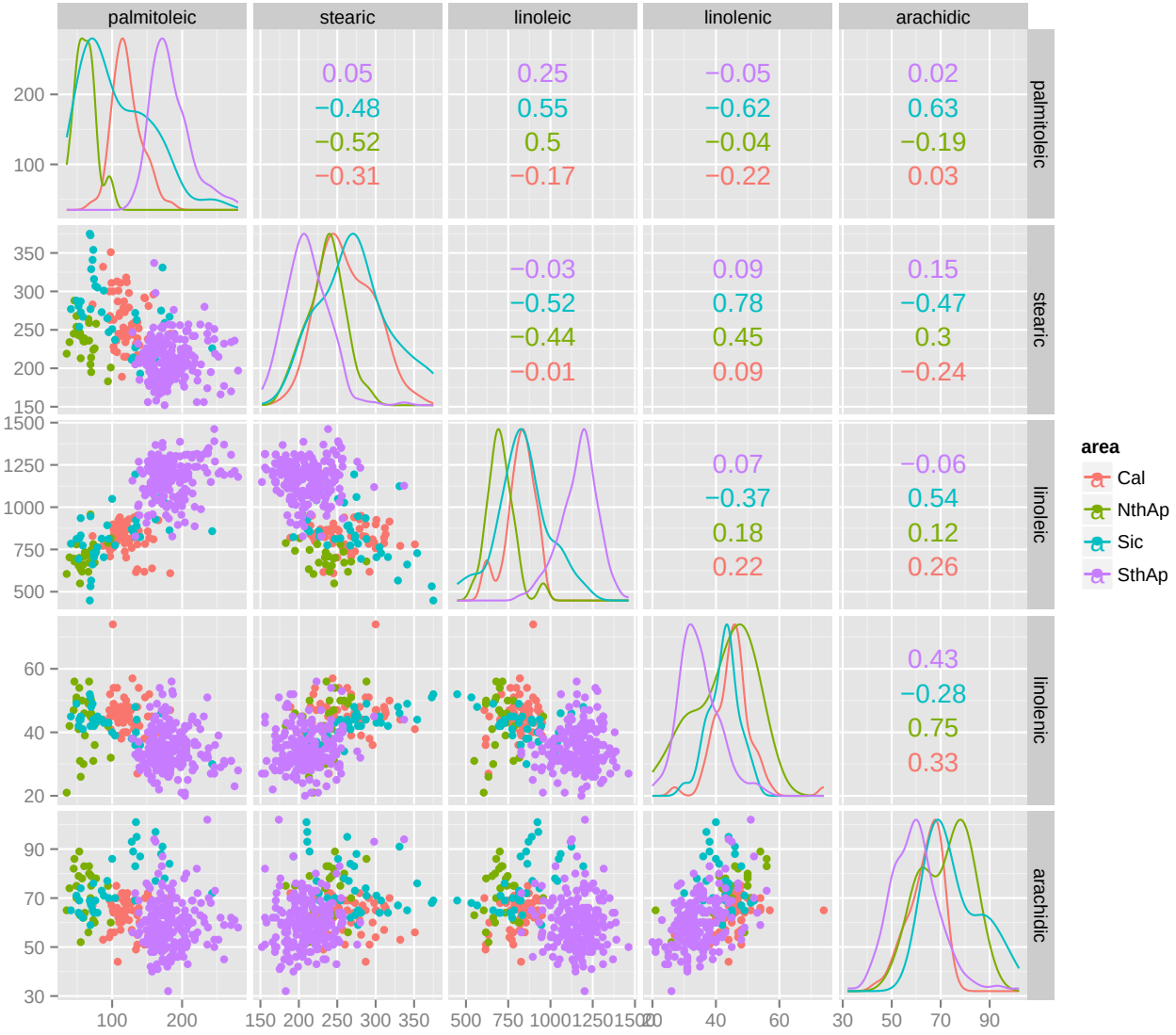
```
##         Cal NthAp Sic SthAp Sum   error
## Cal      22     0   5     1  28 0.21429
## NthAp     1     9   1     1  12 0.25000
## Sic       2     1  14     1  18 0.22222
## SthAp     0     0   4   100 104 0.03846
```

Test error = 0.105

Example: Tree model



Example: A look at the data



Example: Fit a random forest model

```
##
## Call:
##  randomForest(formula = area ~ ., data = olive.sth, importa
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 6.19%
## Confusion matrix:
##       Cal NthAp Sic SthAp class.error
## Cal    54     0   0     2    0.035714
## NthAp   2    22   1     0    0.120000
## Sic     4     3  23     6    0.361111
## SthAp   0     0   2   204    0.009709
```

Example: Think about it

- Error rates: notice anything?
- What were the input parameters?

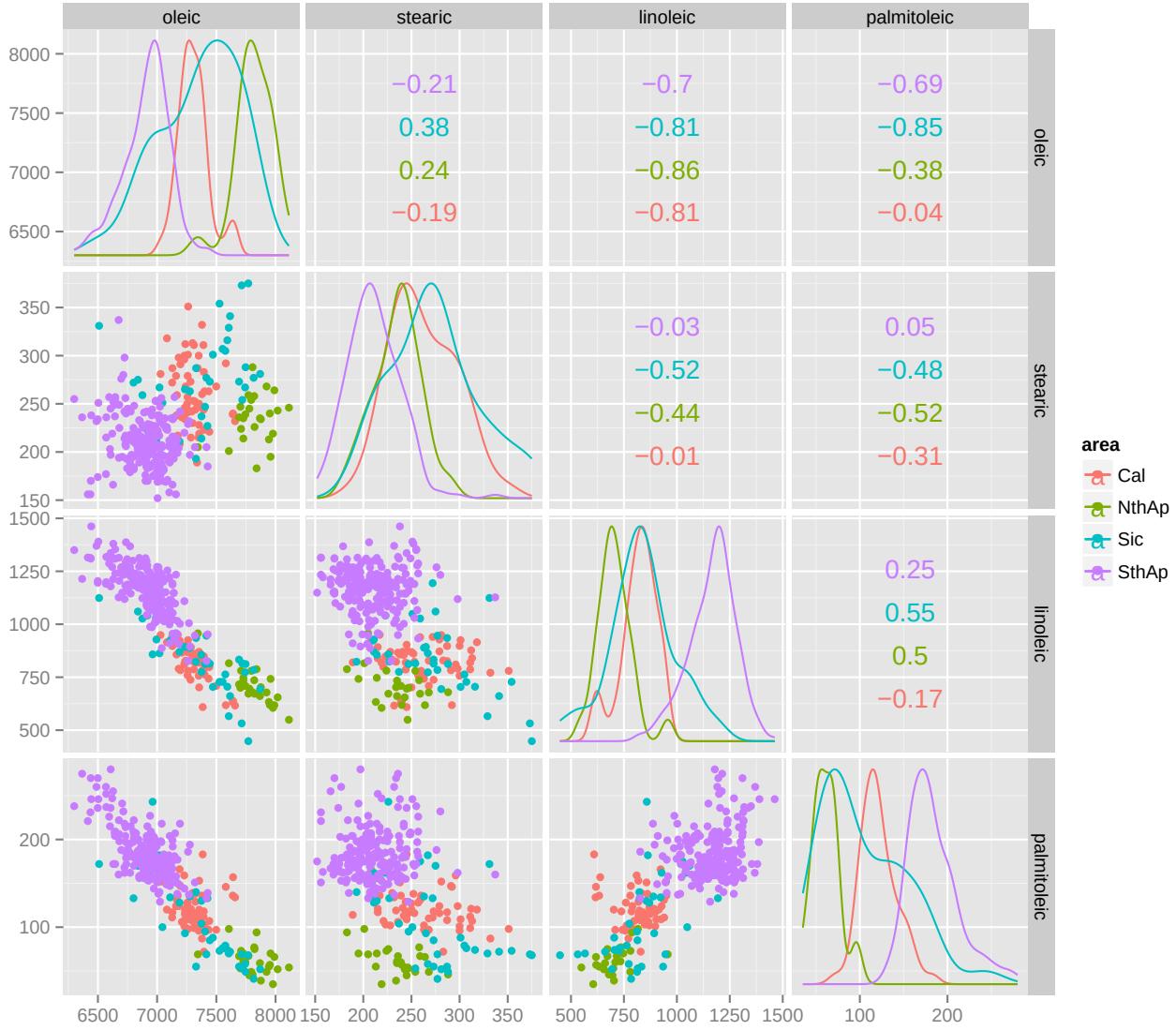
Example: Variable importance, overall

```
##          vars MeanDecreaseAccuracy MeanDecreaseGini
## 1    linoleic                 0.147               37
## 2 palmitoleic                 0.123               36
## 3       oleic                 0.092               33
## 4    palmitic                 0.044               17
## 5     stearic                 0.051               16
## 6   linolenic                 0.040               13
## 7   arachidic                 0.026               12
## 8  eicosenoic                 0.023               11
```

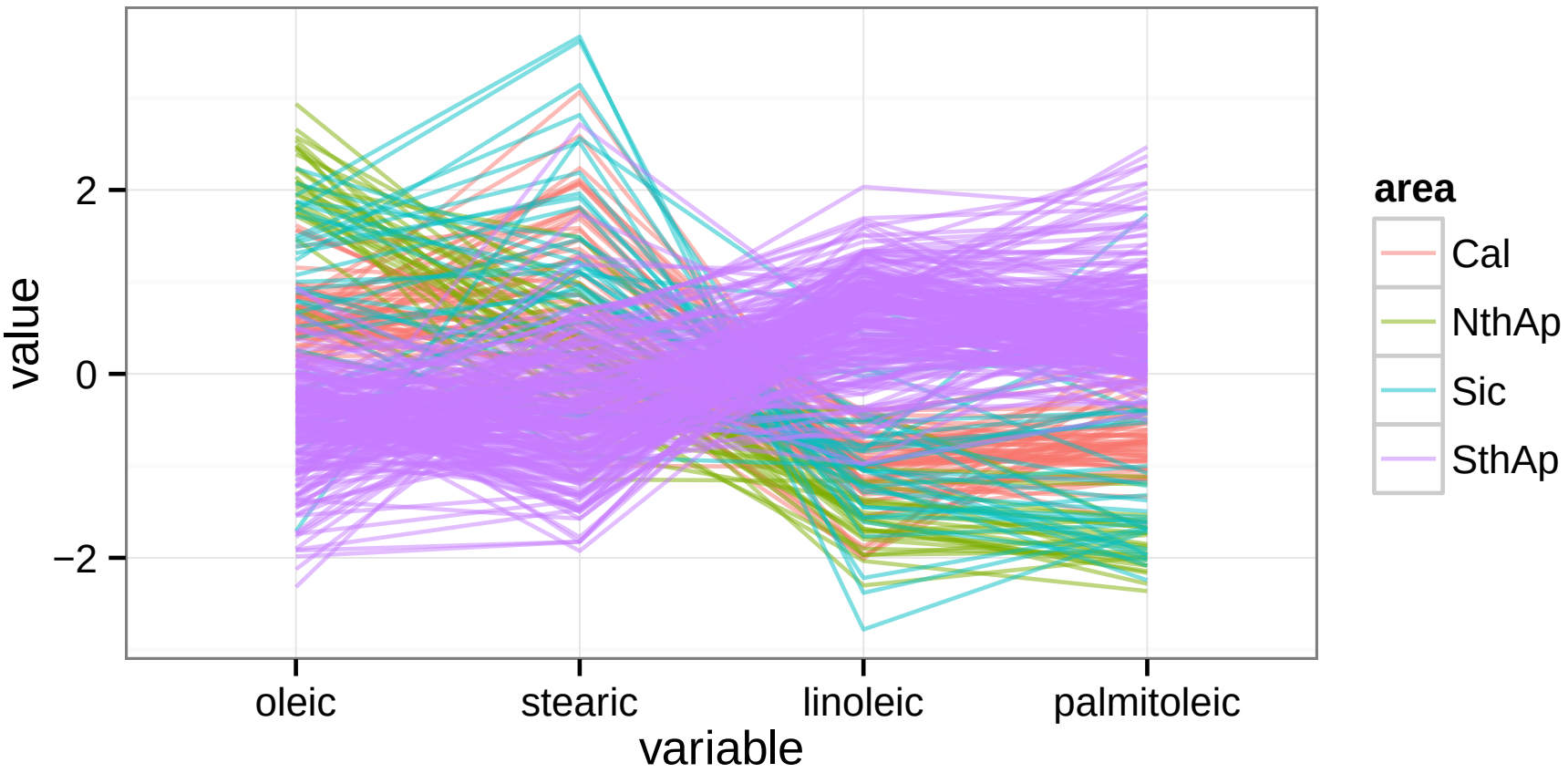
Example: by class

```
##          vars   Cal     NthAp   Sic  SthAp
## 1    linoleic 0.263  0.21312 0.032 0.1292
## 2   linolenic 0.147 -0.00152 0.046 0.0150
## 3       oleic 0.134  0.30168 0.037 0.0657
## 4 palmitoleic 0.111  0.27437 0.124 0.1092
## 5     stearic 0.061  0.00547 0.121 0.0420
## 6    palmitic 0.037  0.31654 0.025 0.0168
## 7   arachidic 0.033  0.05778 0.098 0.0075
## 8  eicosenoic 0.016 -0.00011 0.114 0.0113
```

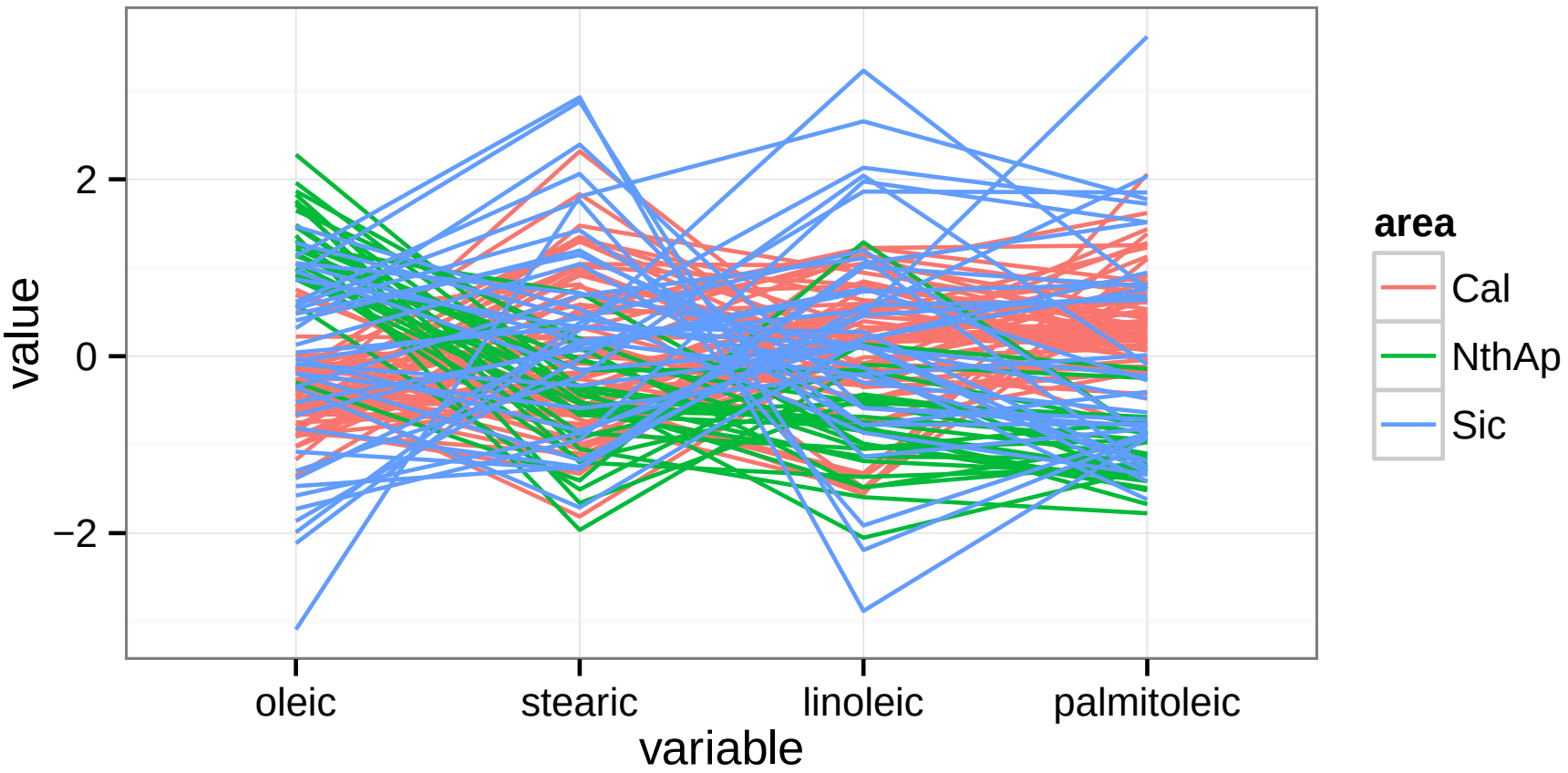
Example: Use to choose vars for scatmat



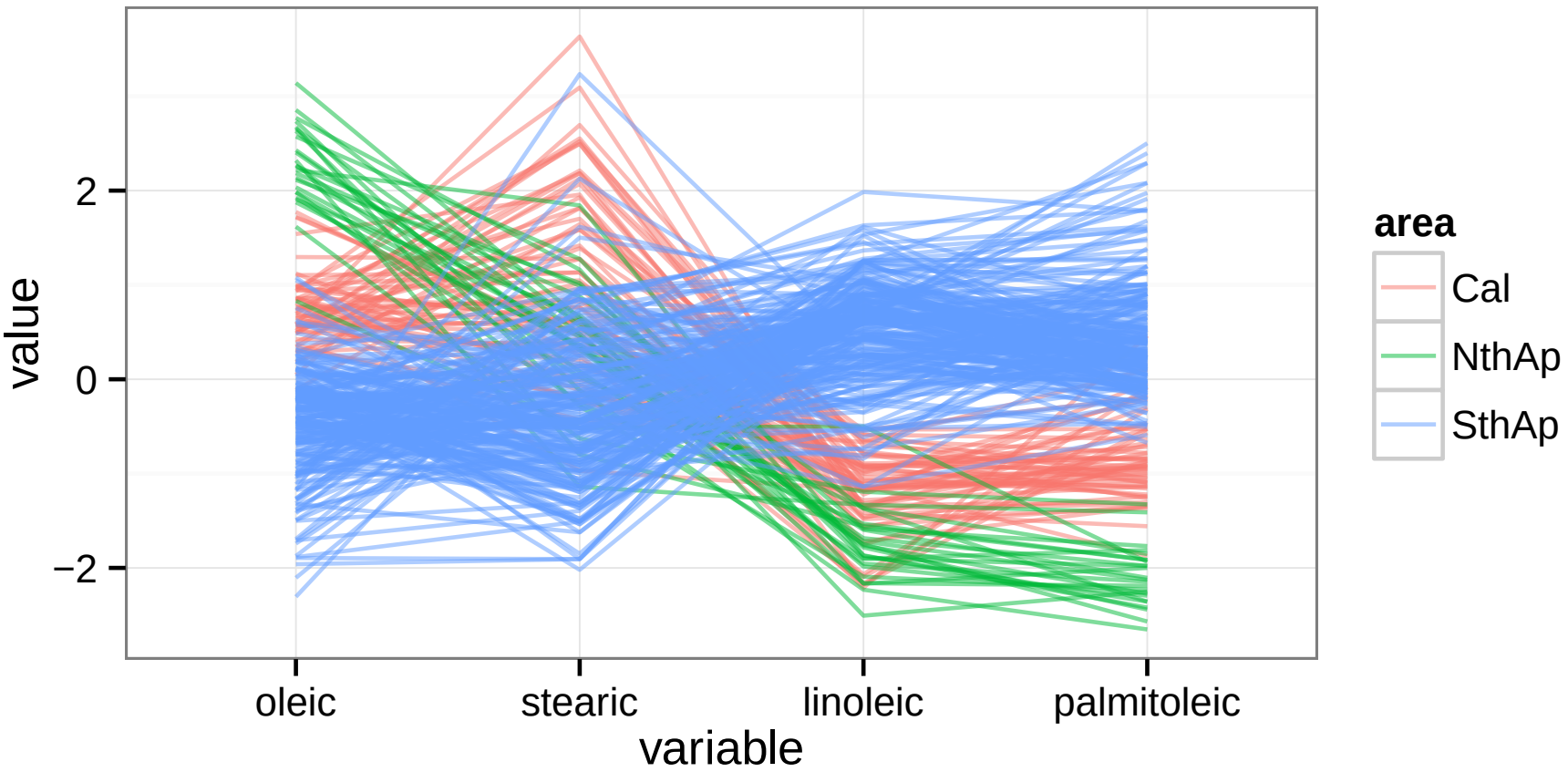
Example: Use to arrange par coords



Example: Without the large group



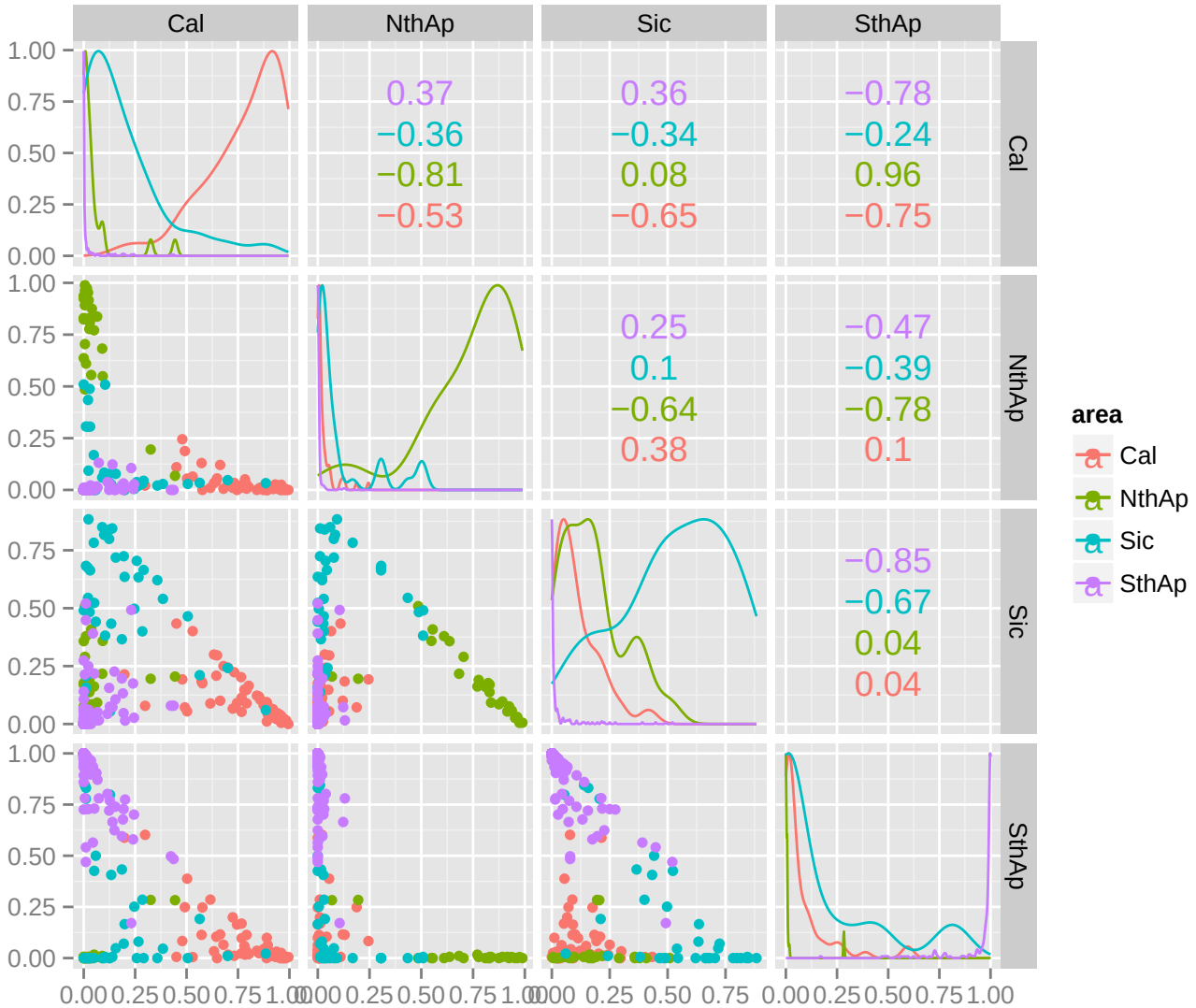
Example: Without the trouble maker class



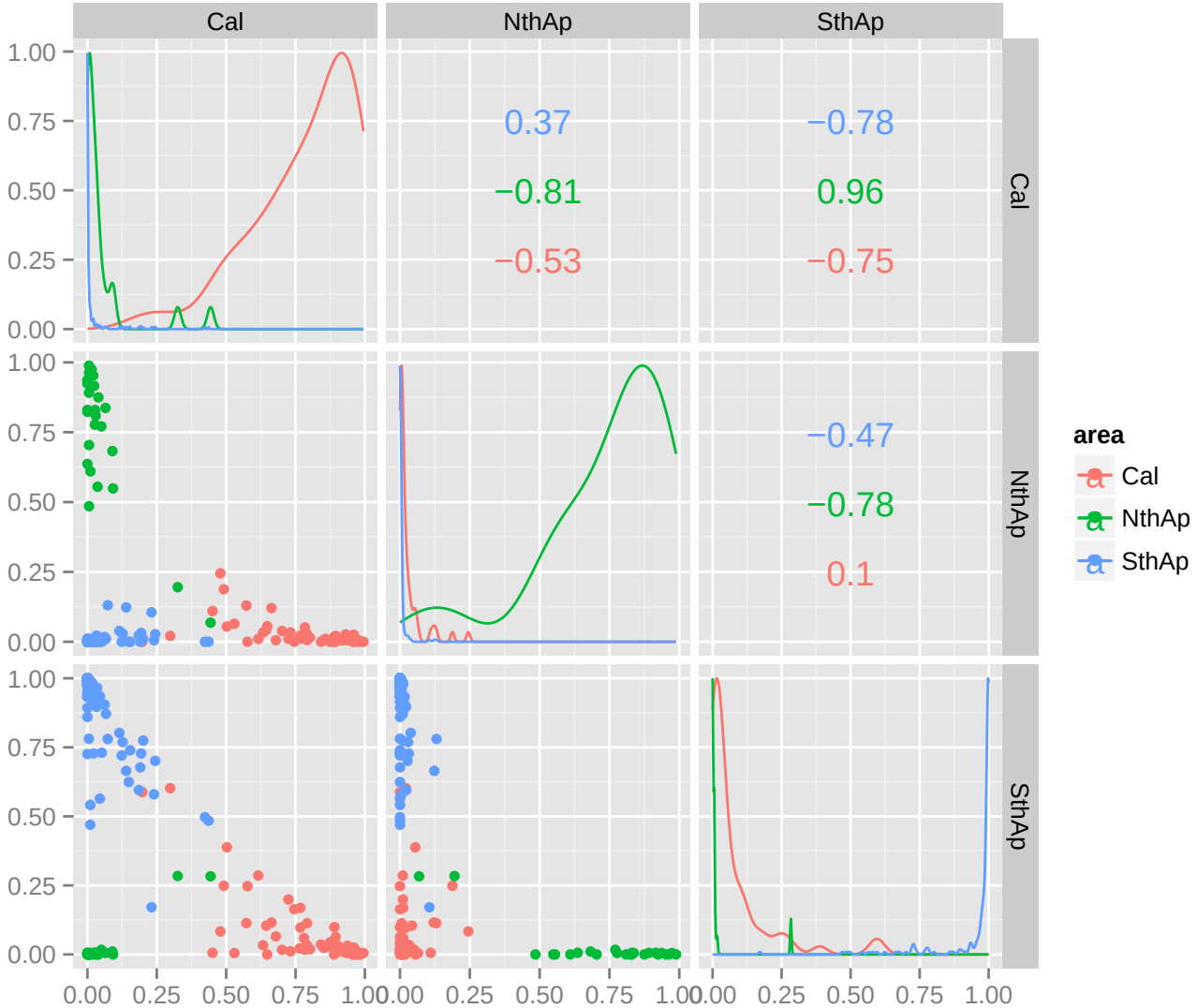
Example: Vote matrix

```
##     Cal  NthAp   Sic SthAp
## 1 0.75 0.0000 0.091 0.164
## 2 0.50 0.0546 0.055 0.388
## 3 0.97 0.0000 0.016 0.011
## 4 0.79 0.0060 0.089 0.113
## 5 0.90 0.0055 0.066 0.033
## 6 0.68 0.0054 0.250 0.065
```

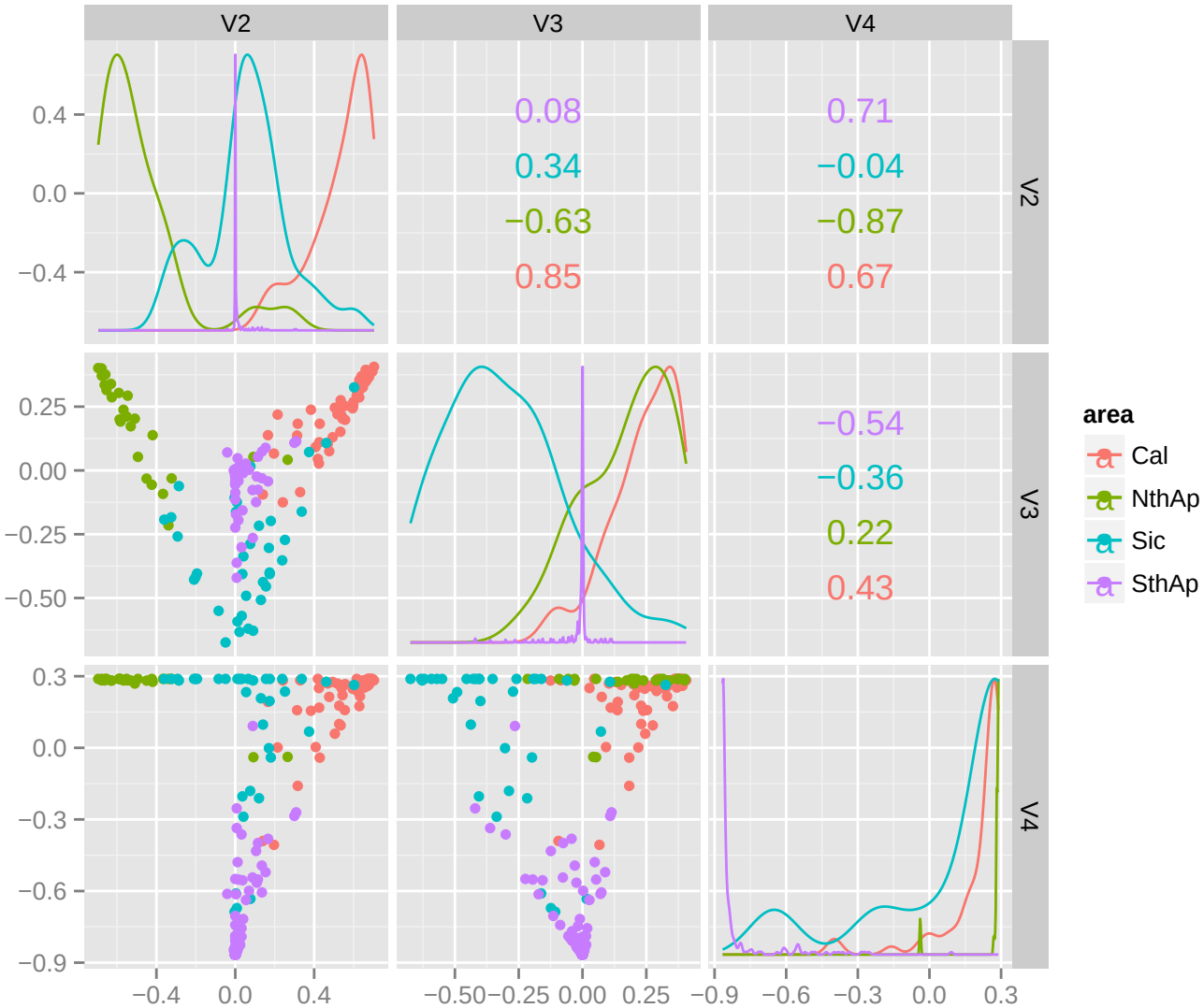
Example: Vote matrix



Example: Vote matrix



Example: Vote matrix



Videos explaining exploring trees and forests

- Trees
- Forests

Proximity matrix

- 323×323 matrix, effectively a distance matrix for all cases from each other
- These distances can be passed to an unsupervised classification, clustering, to examine similarity between cases.
- You would expect cases in different classes to be further from each other, cases within the same class to be close to each other by this metric.
- We will talk about clustering in the next section of the class.

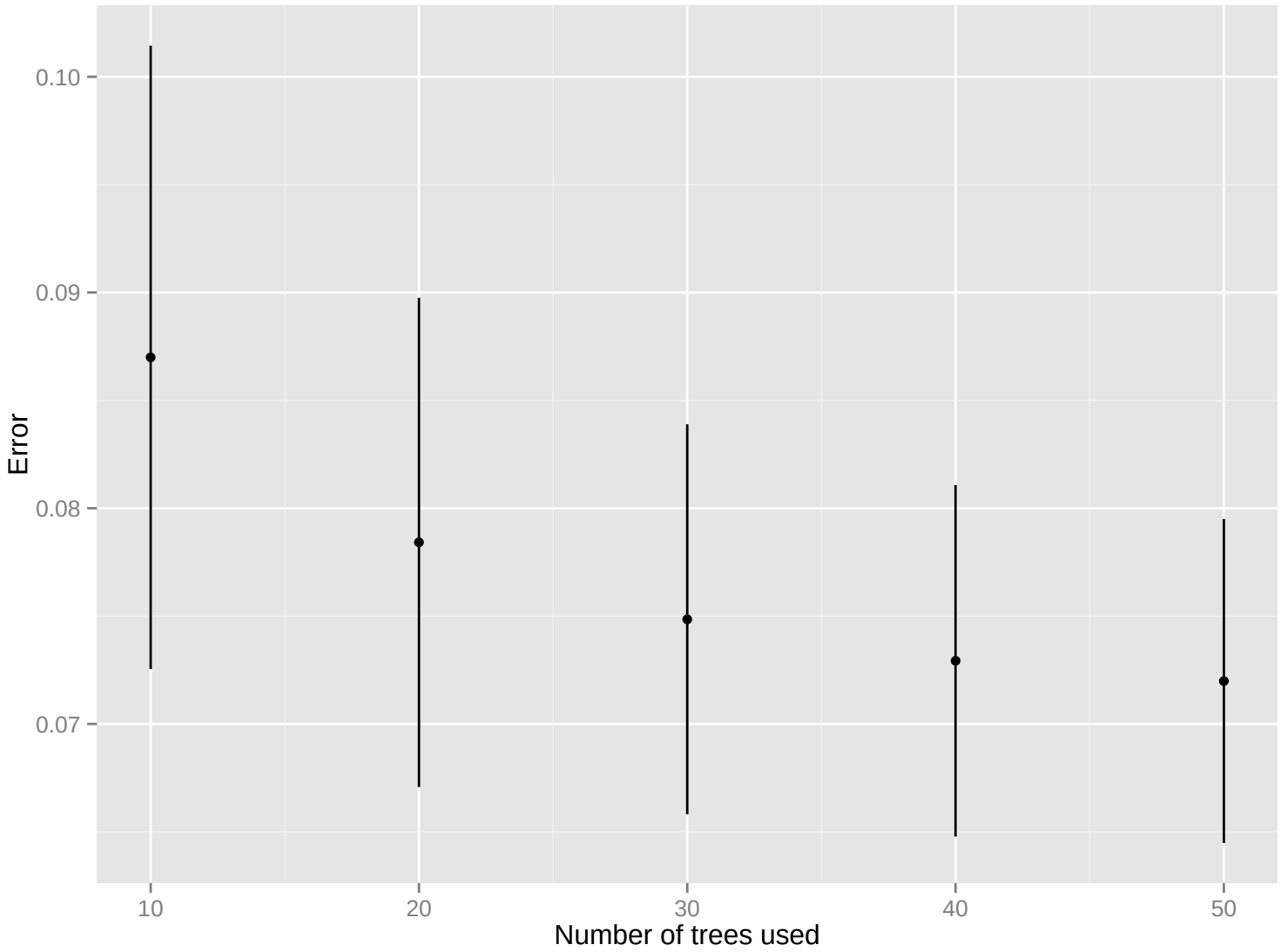
Bagging algorithm

1. Input: $L = \{(x_i, y_i), i = 1, ..., n\}, y_i \in \{1, ..., g\}$
2. Sample: $L^{*b} = \{(x_i^{*b}, y_i^{*b}), i = 1, ..., n\}, b = 1, ..., B$ Sample with replacement, $p_i = 1/n$.
3. Fit classifier C_b to each L^{*b} and predict x_{ib}, y_{ib}.
4. Combine predictions, perhaps by majority rule, class which gets the most votes, to get predicted values for each case.

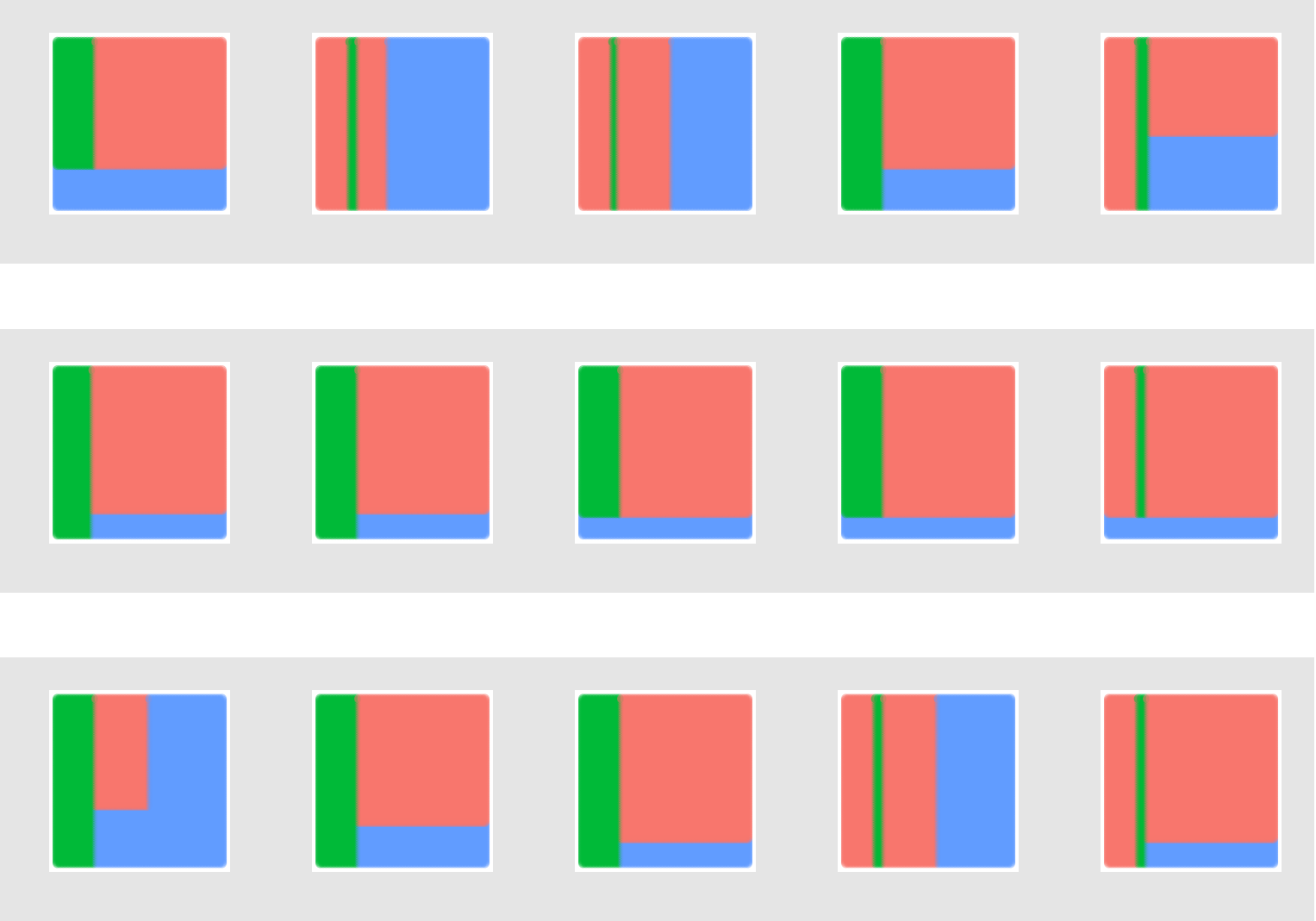
Bagged trees

- Take bootstrap samples of 50% data set, examine the error for the cases left out
- Fit one tree, calculate oob error, and mean/variance of this error for repeating many times
- Fit many trees, combine predictions, calculate error, and mean/variance of error for many repetitions

Bagged trees - 50 trees



Bagged trees - boundaries



Bagged trees - boundaries