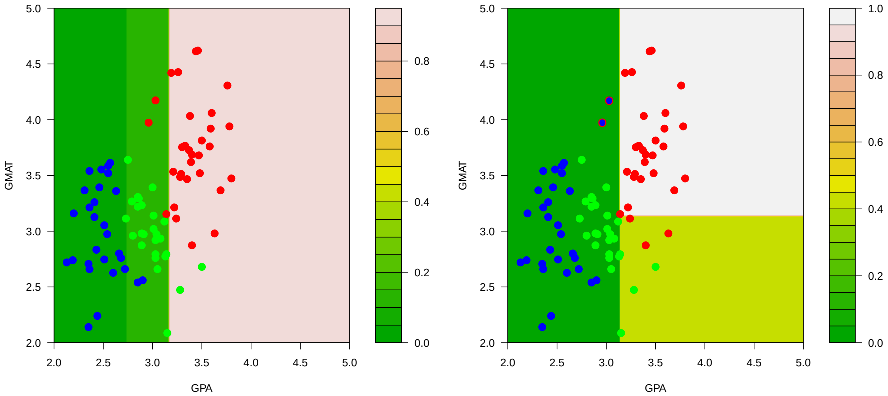
STAT406- Methods of Statistical Learning Lecture 16

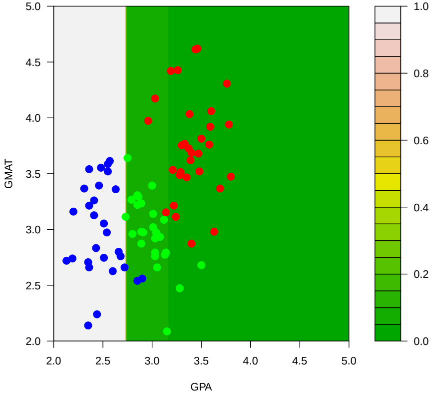
Matias Salibian-Barrera

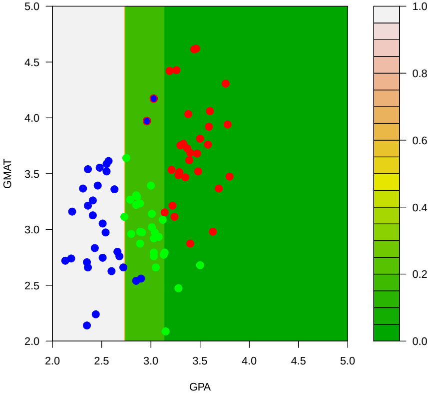
UBC - Sep / Dec 2016

Trees are unstable...

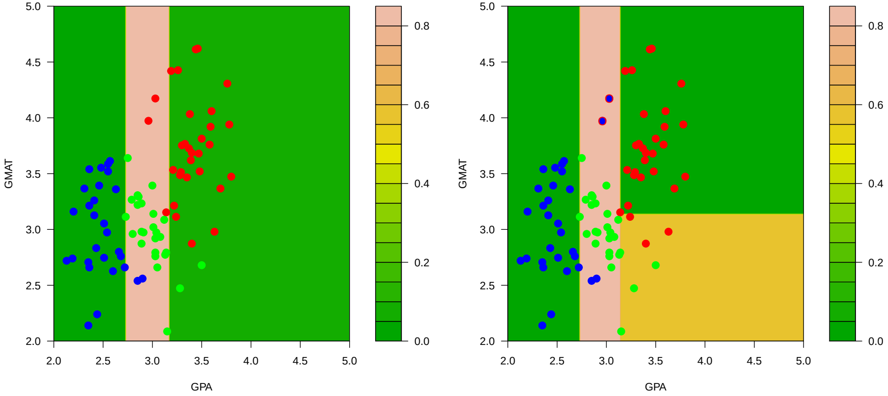


Trees are unstable...

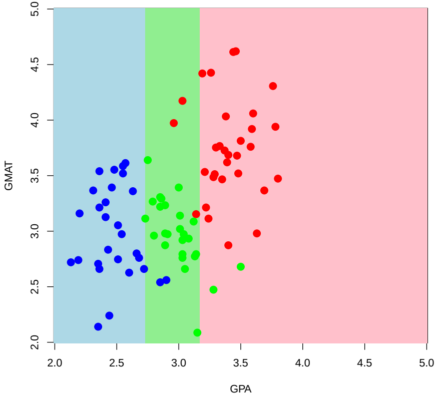


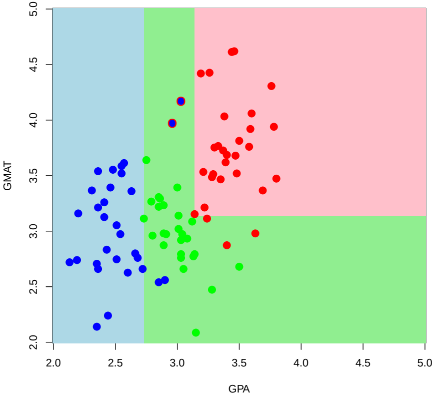


Trees are unstable...



Trees are unstable...





Classification Trees - Bagging

- This problem can be alleviated sometimes using “Bagging” (Bootstrap aggregation)
- It is a general principle, applies to any classifier / estimator
- Consider the predicted class $\hat{g}(\mathbf{X})$ or predicted probabilities $\hat{P}(g|\mathbf{X})$

$$\hat{f}(\mathbf{X}) = \left\{ \begin{array}{c} \left(\hat{P}(g_1|\mathbf{X}), \ldots, \hat{P}(g_K|\mathbf{X}) \right) \\ \\ \arg\max_g \hat{P}(g|\mathbf{X}) \end{array} \right\}$$

Classification Trees - Bagging

- If we had several independent samples, we could obtain a more stable (less variable) $\hat{f}(\mathbf{X})$ by using the average over the samples.

- Using our sample distribution as an estimator of the population distribution, the bootstrap simulates independent samples by randomly drawing samples from our data

Classification Trees - Bagging

- We can obtain a "large" number of trees and have them "vote" on the classification of future observations, or average their posterior probabilities estimates $\hat{P}\left(g_j \middle| \mathbf{X}\right)$

Classification Trees - Bagging

- Our aggregated classifier is

$$\bar{f}(\mathbf{X}) = \left\{ \begin{array}{c} \left(\bar{P}(g_1|\mathbf{X}), \ldots, \bar{P}(g_K|\mathbf{X})\right) \\ \\ \arg\max\,(n_1, n_2, \ldots, n_K) \end{array} \right\}$$

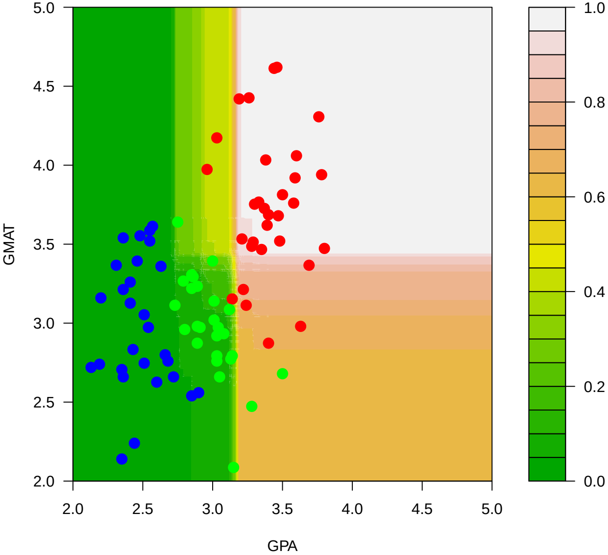
where $\left(\bar{P}(g_1|\mathbf{X}), \ldots, \bar{P}(g_K|\mathbf{X})\right)$ are averaged posterior probabilities over the boostrap samples;

$(n_1, n_2, \ldots, n_K)$ are the number of times each class was selected and $n_1 + n_2 + \cdots + n_K =$ number of bootstrap samples

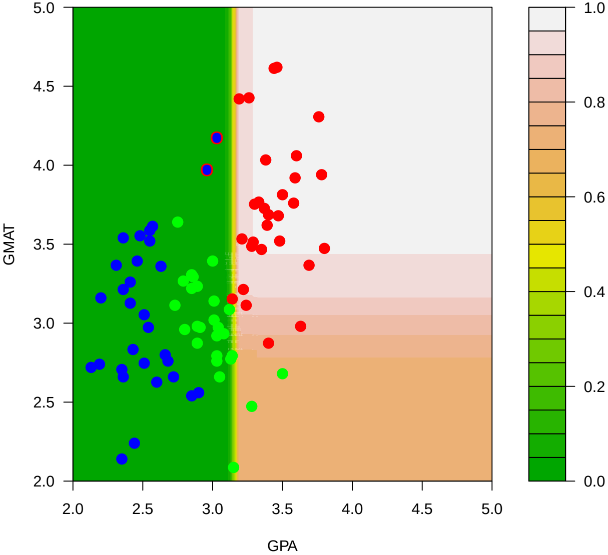
Example

```
Example
```

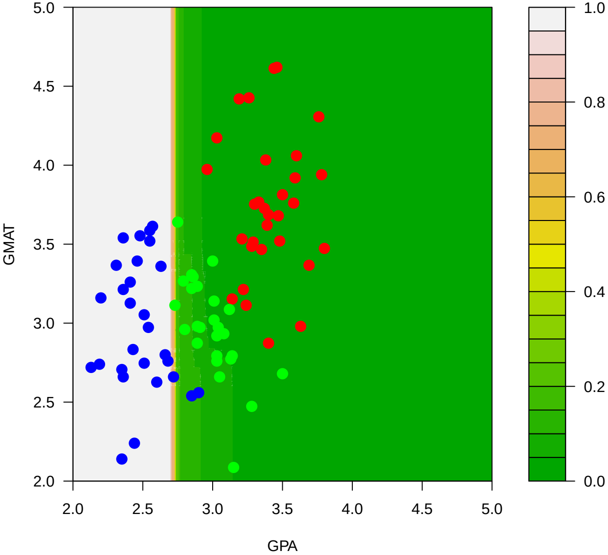
Bagged trees -original data



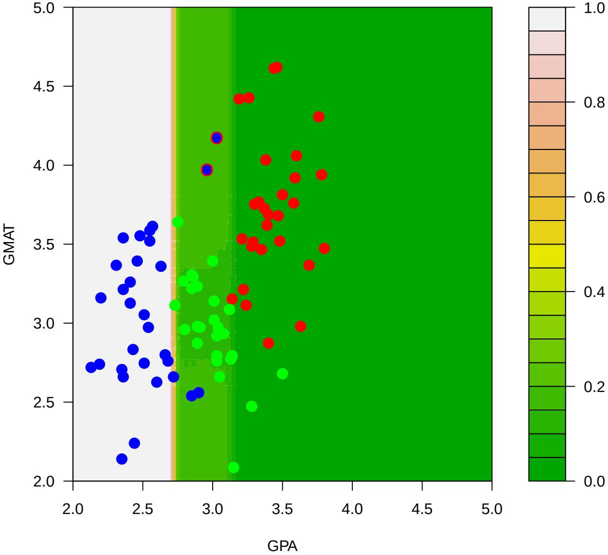
Bagged trees -modified data



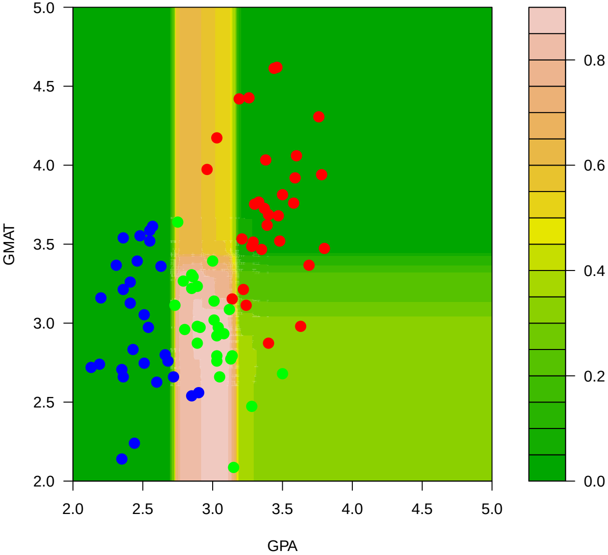
Bagged trees -original data



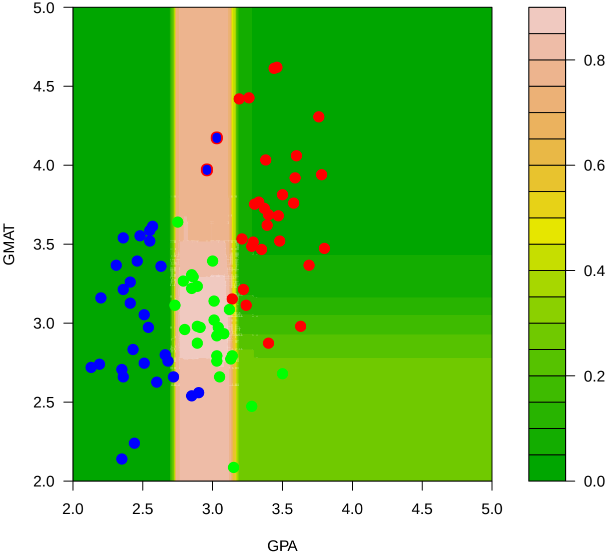
Bagged trees - modified data



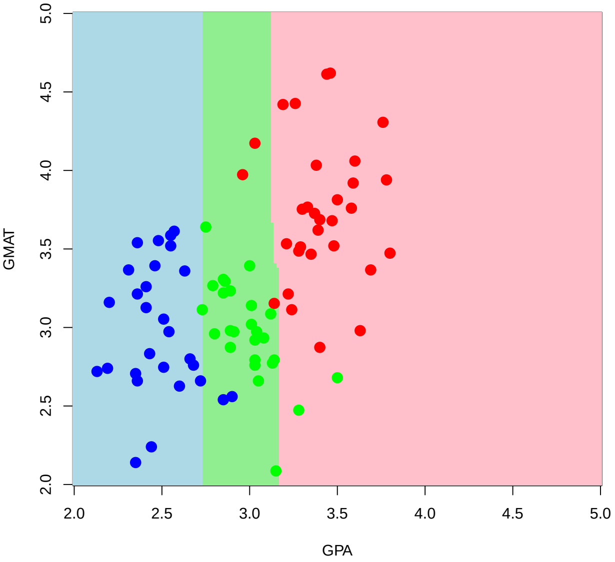
Bagged trees -original data



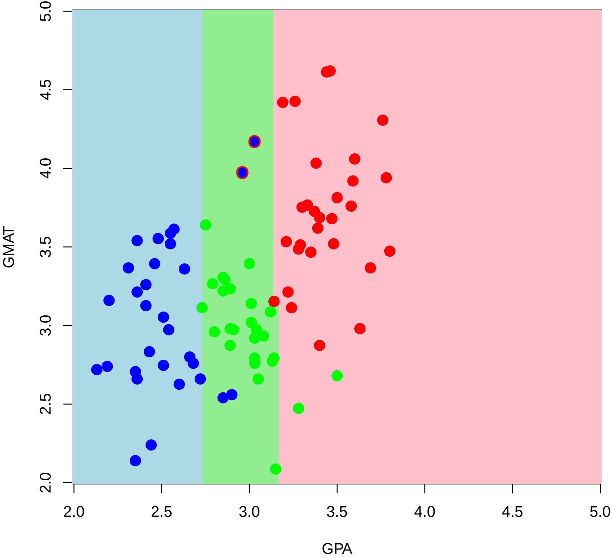
Bagged trees - modified data



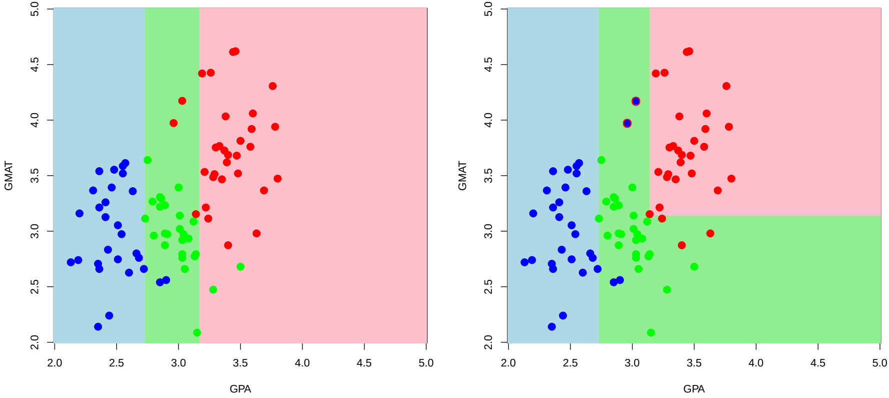
Bagged trees -original data



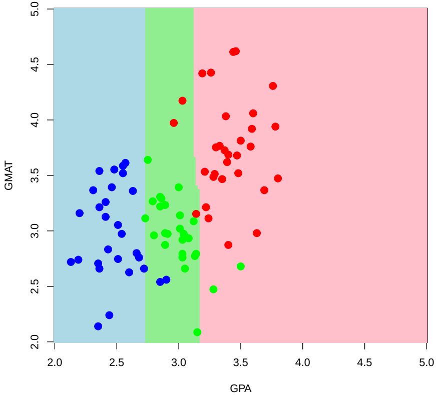
Bagged trees - modified data

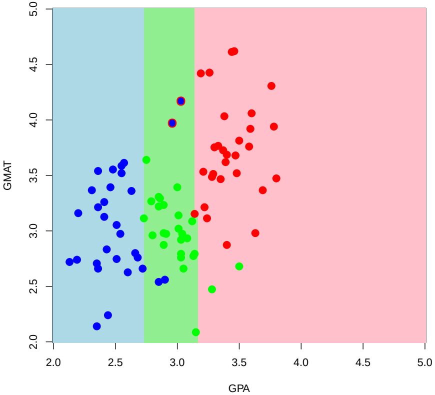


Trees are unstable...



Bagging helps





Random forests

- Bagging - averaging identically distributed trees (which may be correlated)

- Random forests - making the “bagged” trees less correlated
- The bootstrapped trees are de-correlated by making them use different features for the splits

Random forests

(1) `for(b in 1:B)`

  (a) Draw a bootstrap sample from the training data

  (b) Grow a "random forest tree" as follows: for each terminal node:

    (i) Randomly select `m` features

    (ii) Pick the best split among these

    (iii) Split the node into two children

(2) Return the ensemble of trees $(T_b)_{1 \leq b \leq B}$

Random forests

- With these B trees we can do predictions
- Given a new point $\mathbf{x}$, for regression we use

$$\hat{f}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} T_b(\mathbf{x})$$

For classification:

$$\hat{f}(\mathbf{x}) = \text{majority vote among} \Big\{ T_b(\mathbf{x}) \,, \; 1 \le b \le B \Big\}$$