

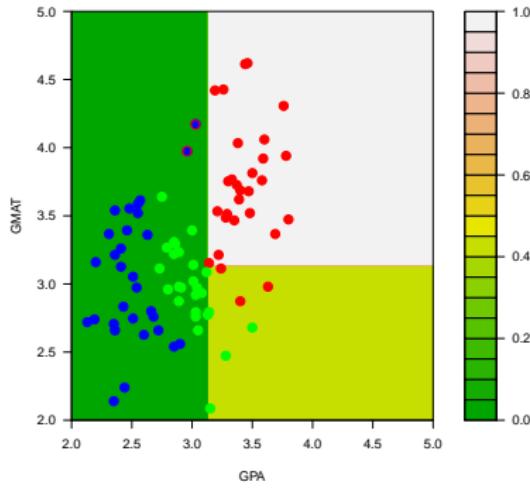
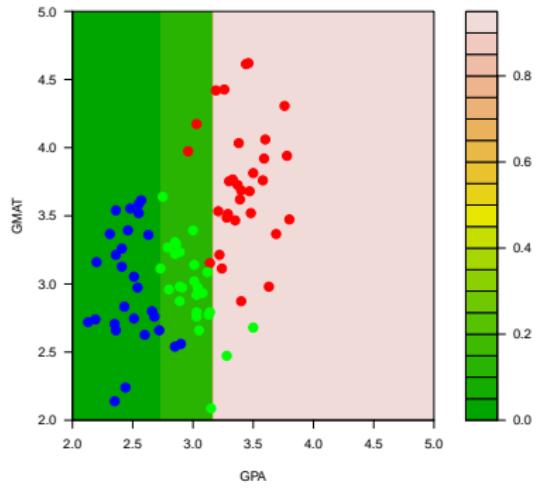
STAT406- Methods of Statistical Learning

Lecture 16

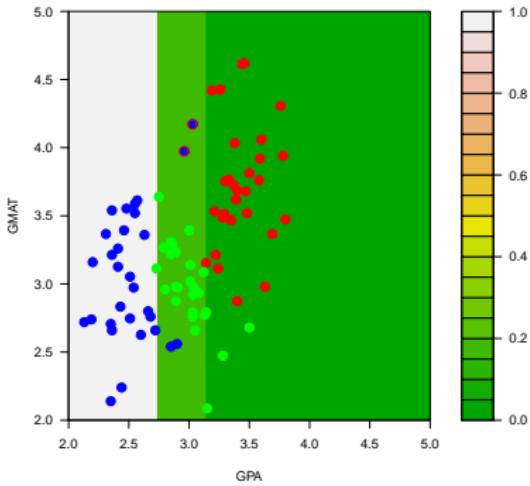
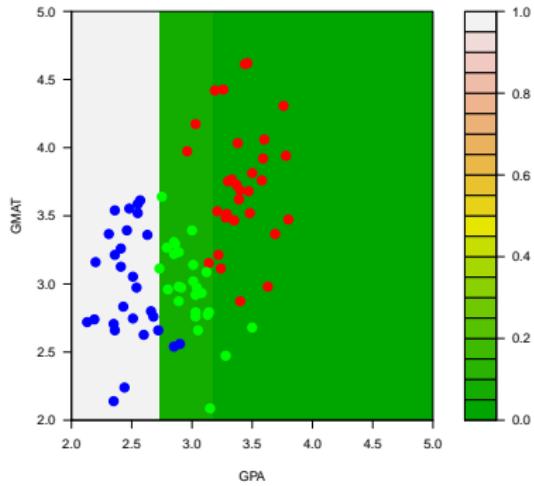
Matias Salibian-Barrera

UBC - Sep / Dec 2017

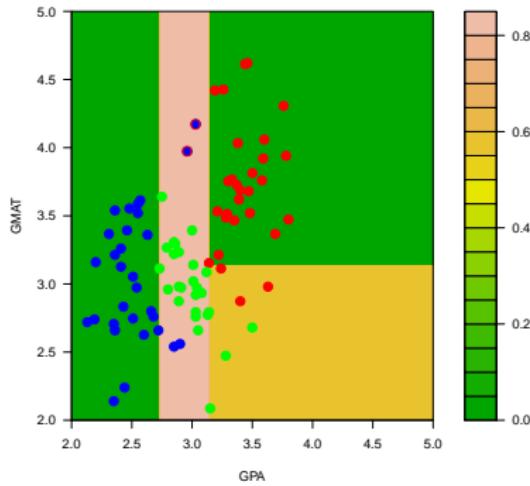
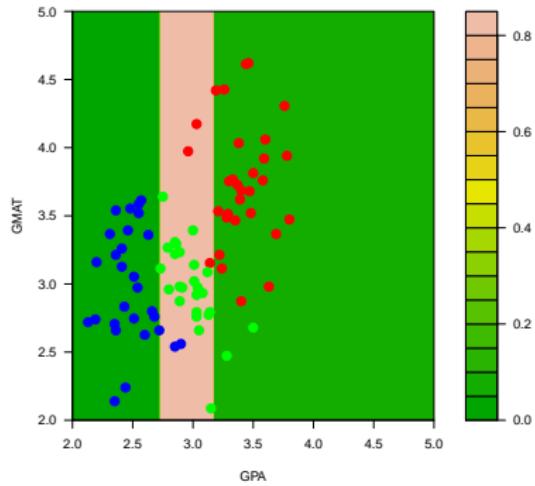
Trees are unstable...



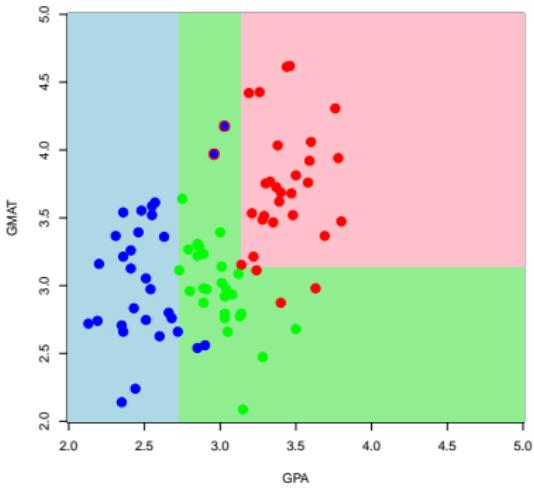
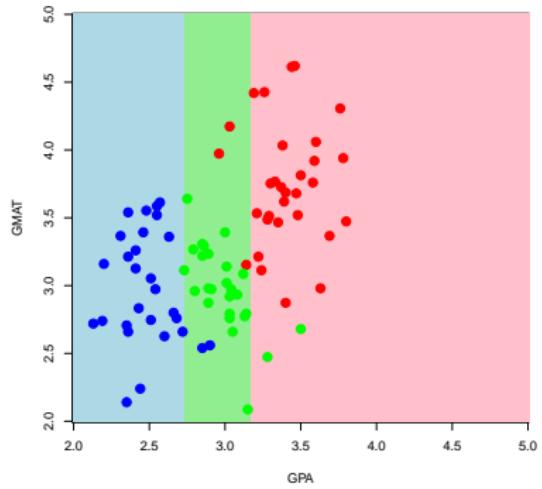
Trees are unstable...



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
- But it's not always equally effective (see the Midterm)

Classification Trees - Bagging

- Consider the predicted class $\hat{g}(\mathbf{X})$ or predicted probabilities $\hat{P}(g|\mathbf{X})$

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

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 **conditional probabilities estimates** $\hat{P}(g_j | \mathbf{x})$

Classification Trees - Bagging

- Our aggregated classifier is

$$\bar{f}(\mathbf{X}) = \left\{ \begin{array}{l} (\bar{P}(g_1|\mathbf{X}), \dots, \bar{P}(g_K|\mathbf{X})) \\ \arg \max (\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_K) \end{array} \right\}$$

where $(\bar{P}(g_1|\mathbf{X}), \dots, \bar{P}(g_K|\mathbf{X}))$ are averaged conditional probabilities over the bootstrap samples;

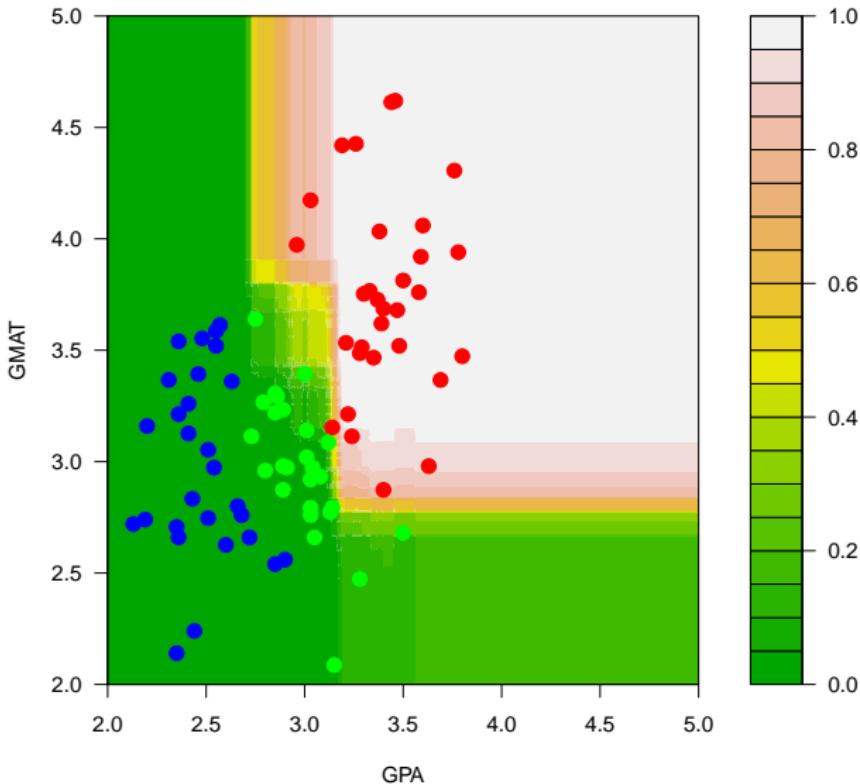
$(\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_K)$ are the number of times each class was selected and

$\mathbf{n}_1 + \mathbf{n}_2 + \dots + \mathbf{n}_K =$
number of bootstrap samples

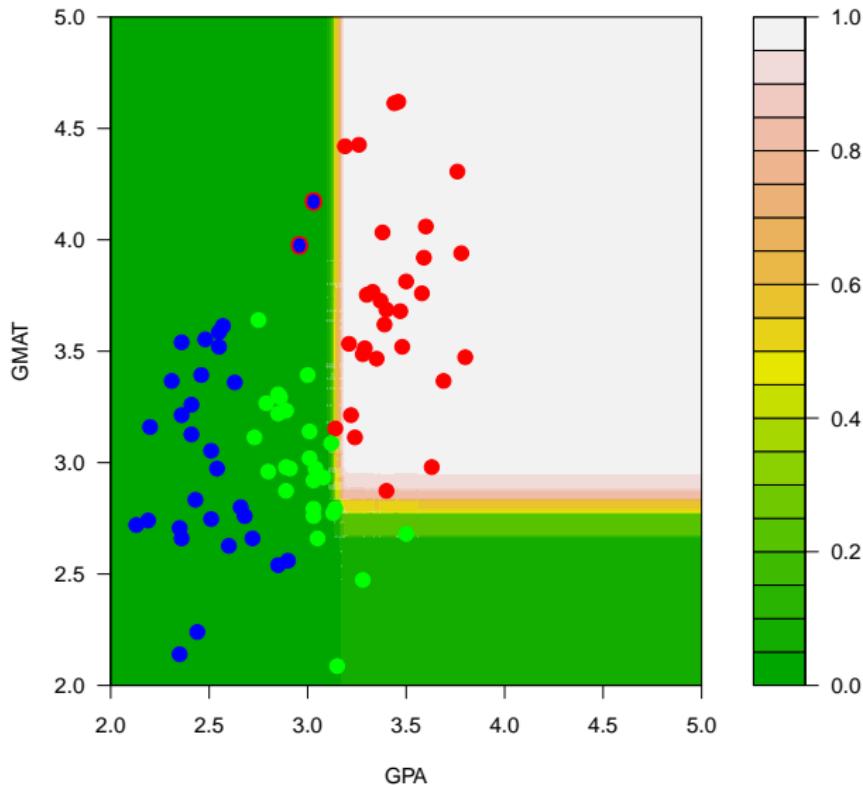
Example

Example

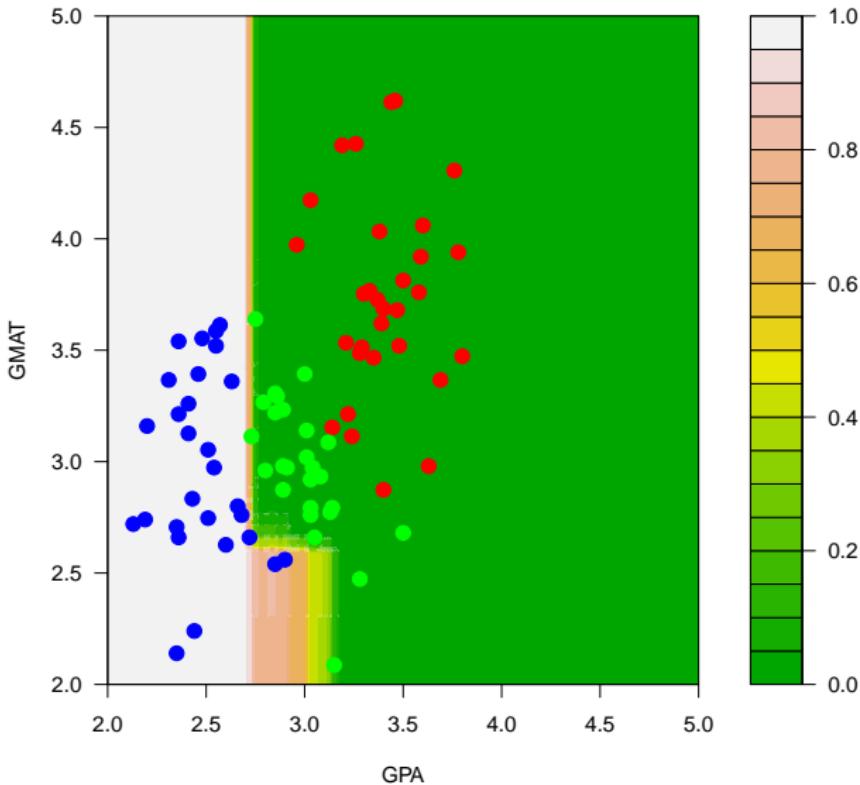
Bagged trees -original data



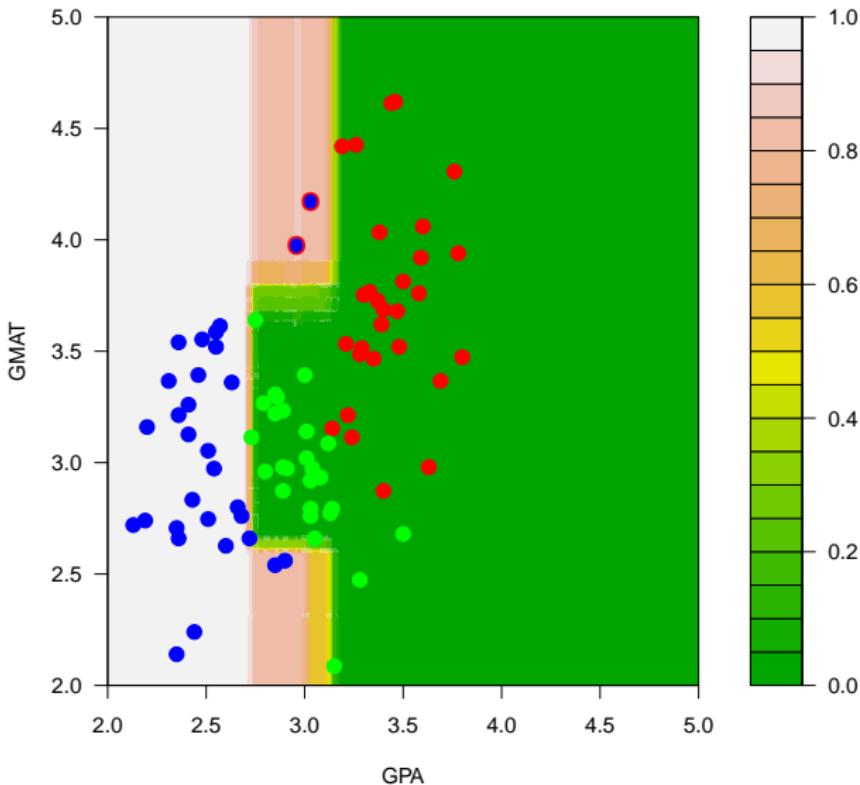
Bagged trees -modified data



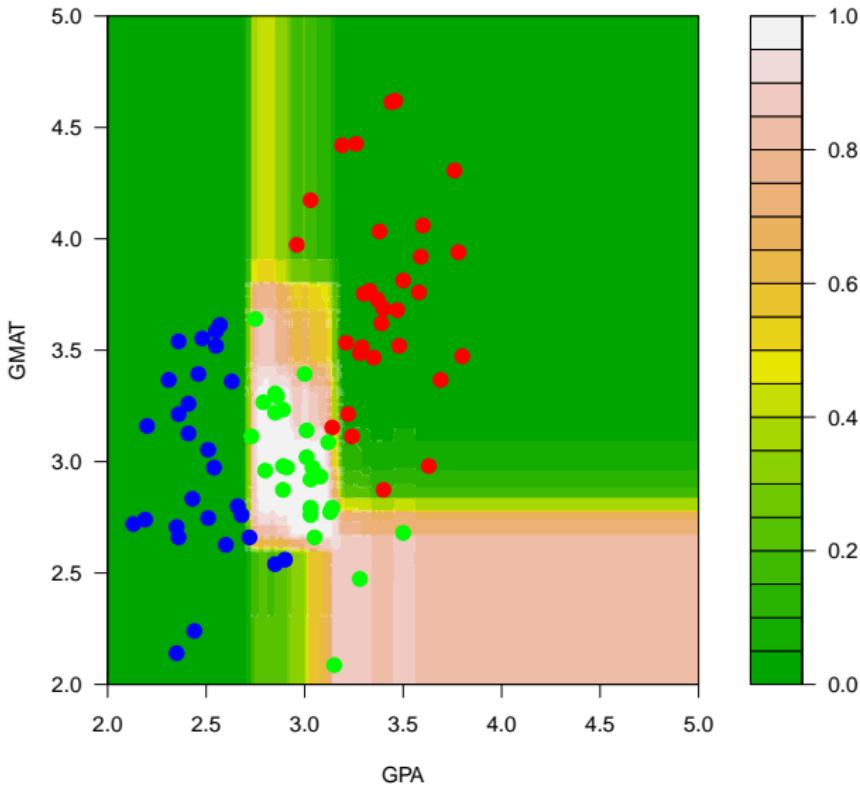
Bagged trees -original data



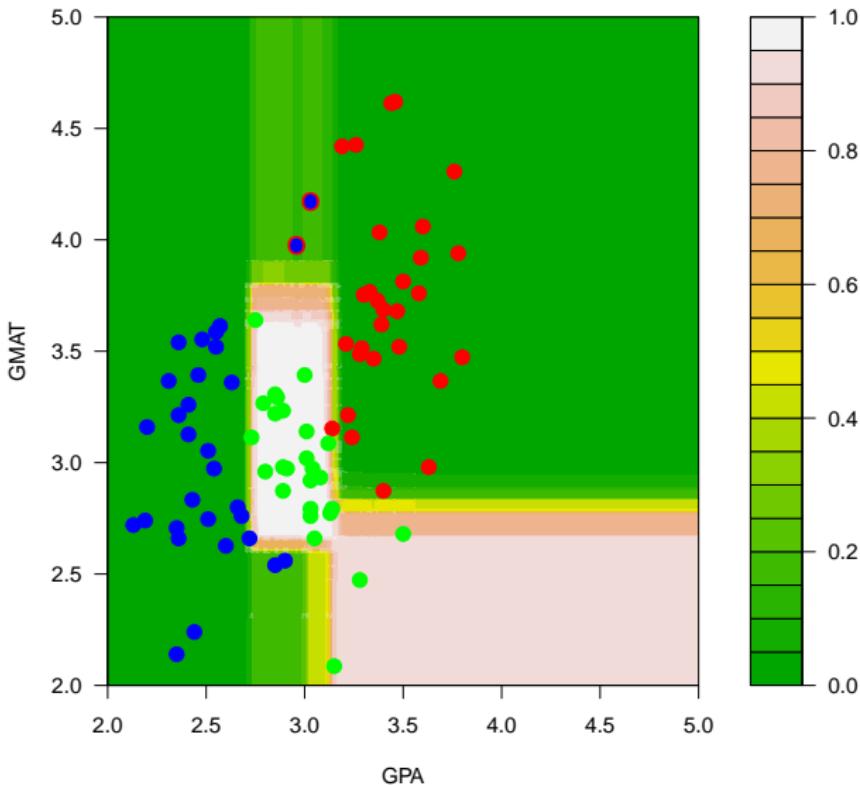
Bagged trees - modified data



Bagged trees -original data



Bagged trees - modified data



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
- (c) Repeat (b) to grow a (very very) large tree

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

Random forests

- 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:

$$\hat{f}(\mathbf{x}) = \text{majority vote among} \left\{ T_b(\mathbf{x}), 1 \leq b \leq B \right\}$$

Q: why not average conditional prob's?

Out-of-bag error estimates

- Each bagged tree is trained on a bootstrap sample
- Predict the observations not in the bootstrap sample with that tree
- One will have “about” $B/3$ predictions for each point in the training set
- These can be used to estimate the prediction error (classification error rate) without having to use CV

Out-of-bag error estimates

- For each training observation (y_i, \mathbf{x}_i) , obtain a prediction using only those trees in which (y_i, \mathbf{x}_i) was **NOT** used
- In other words, let \mathcal{I}_i the set of trees (bootstrap samples) where (y_i, \mathbf{x}_i) does not appear, then

$$\hat{y}_i = \frac{1}{|\mathcal{I}_i|} \sum_{j \in \mathcal{I}} T_j(\mathbf{x}_i)$$

Random forests

- This error estimate can be computed at the same time as the trees are being built
- When this error estimate is stabilized we can stop adding trees to the ensemble

Example

Example