Computational Data Analysis

Machine Learning

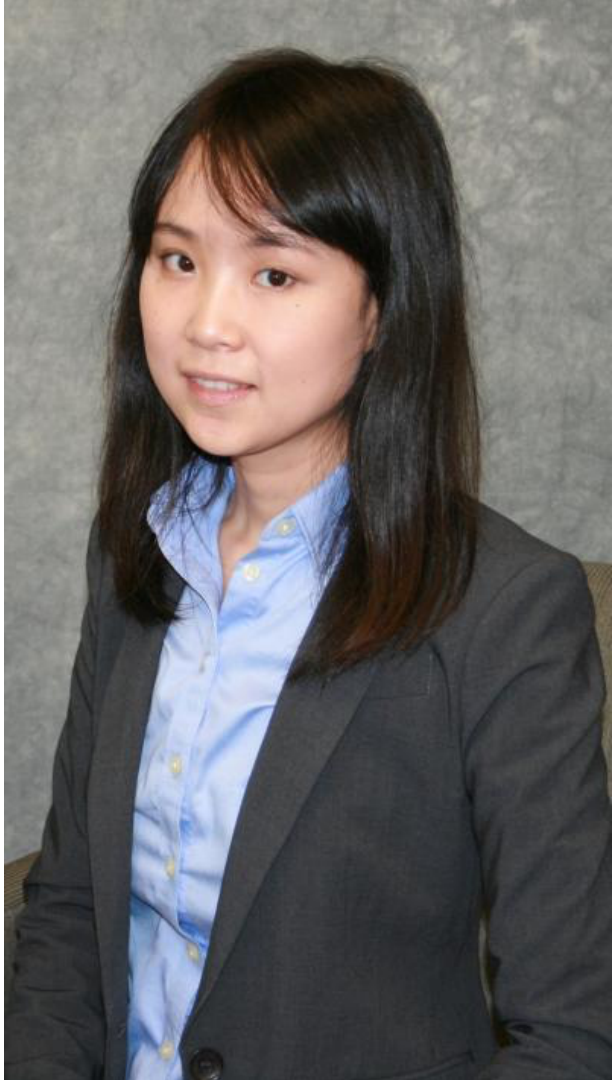
Yao Xie, Ph.D.

Associate Professor

Harold R. and Mary Anne Nash Early Career Professor

H. Milton Stewart School of Industrial and Systems Engineering

Decision Tree and Random Forest



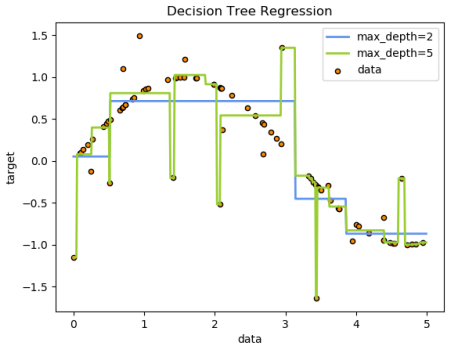
Decision tree

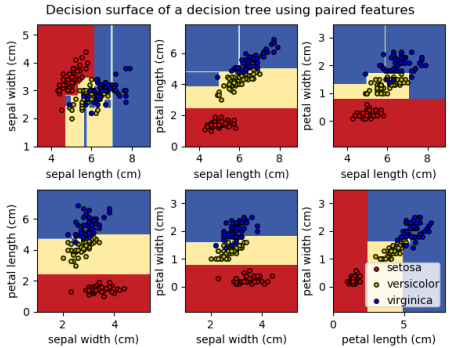
- Decision Trees (DTs) are a non-parametric supervised learning method used for **classification** and **regression**.
- The goal is to predict a target variable by learning simple decision rules inferred from the data features
- **Classification And Regression Tree (CART)** (Breiman et al. 1984)

- Simple method and easy to interpret

- Can be noisy and overfit to data

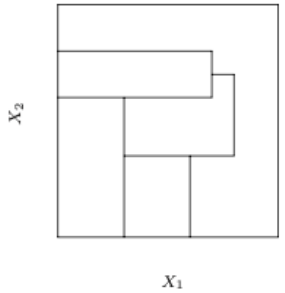
Reference: Chapter 9.2 in ”Elements of Statistical Learning”.

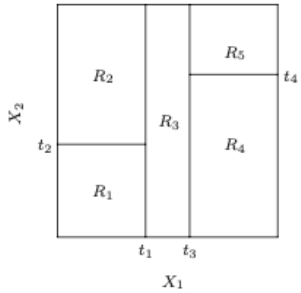


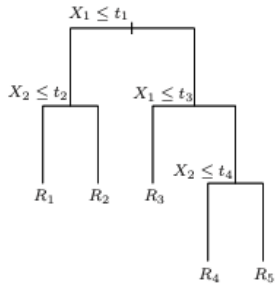


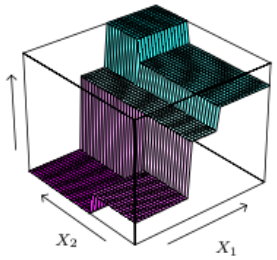
Classification and Regression Tree (CART)

- Partition the feature space into a set of rectangles
- Decision rule can be represented by a tree (binary tree most common)
- Regression tree: Fit a simple model (e.g., a constant) for each rectangle
- Classification tree: Majority vote for samples in the rectangle





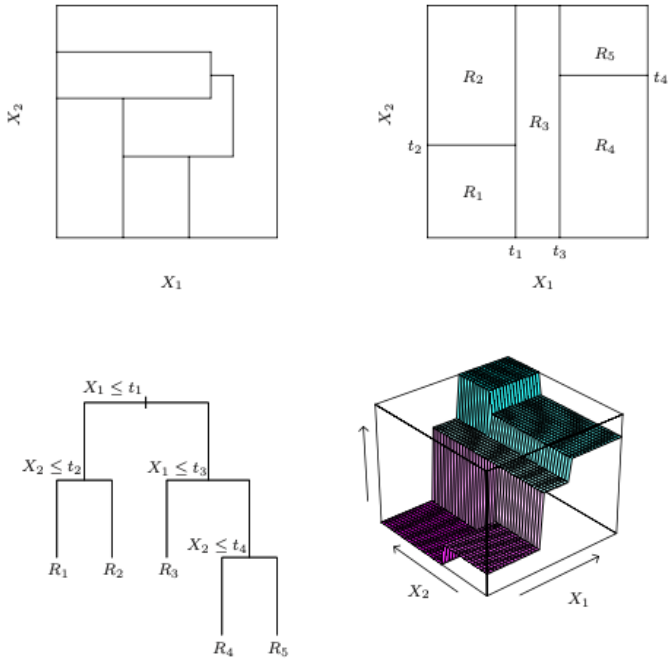




Regression tree:

Simple example

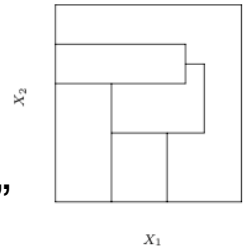
- Restrict attention to recursive binary partitions
- First split the space into two regions and model the response by the mean of Y in each region.
- Choose the variable and split-point to achieve the best fit.
- Then one or both of these regions are split into two more regions
- This process is continued, until some stopping rule is applied.

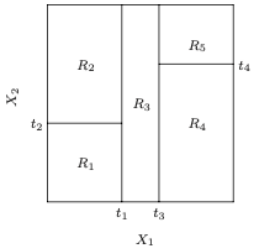


- Final regression model

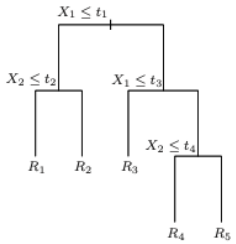
$$\hat{f}(X) = \sum_{j=1}^{5} c_j \, \mathbb{I}\{(X_1, X_2) \in R_j\}$$

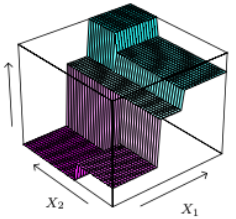
Regression tree





- Greedy approach: Constructing decision trees "top-down"
- Start with all the samples
- Choose a variable at each step that best split the region
  - Splitting a variable j
  - Splitting position s
  - Define half-spaces





$R_1(j,s) = \{X|X_j \le s\}$ and $R_2(j,s) = \{X|X_j > s\}$

Solve the problem for optimal splitting and "regression"

$$\min_{j,s} \left[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2\right]$$

Once (j,s) have decided, optimal prediction is simple

$\hat{c}_1 = \text{ave}(y_i|x_i \in R_1(j,s))$, $\hat{c}_2 = \text{ave}(y_i|x_i \in R_2(j,s))$

Tree pruning

- How large should a tree be?
  - Large tree may overfit data;
  - Small tree underfits (does not capture important structure)
- Determining tree depth by controlling the **data-fit complexity** tradeoff:

Grow a large tree, stop when a minimum node size has reached, then performing pruning by minimizing the cost function

$$C_\alpha(S) = \sum_{j=1}^{|J|} \sum_{x_i \in R_j} \left(y_i - \hat{c}_j\right)^2 + \alpha|J|$$

Measure of data-fitting error for the j-th node

Number of terminal nodes in tree J

($\alpha > 0$: regularization parameter)

Classification tree

- Classification outcome takes value $1,2,\ldots,K$
- Need different criterion for splitting node and pruning tree

$$\hat{p}_{jk} = \frac{1}{N_j} \sum_{x \in R_j} \mathbb{I}(y_i = k)$$

$$k(j) = \arg\max_k \hat{p}_{jk}$$

} For classification

- Tree pruning

$$C_\alpha(J) = \sum_{j=1}^{|J|} N_j Q_j(J) + \alpha |J|$$

Measure of error for the j-th node

Different measure of errors $Q_m(J)$

- Misclassification error

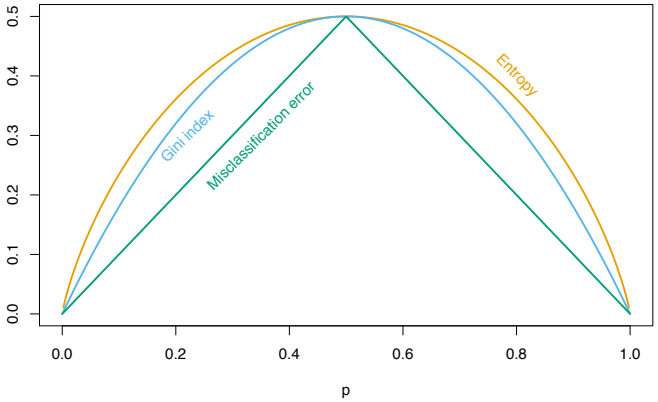
$$\frac{1}{N_j}\sum_{i\in R_j}\mathbb{I}(y_i \neq k(j)) = 1-\hat{p}_{jk}$$

- Gini index

$$\sum_{k=1}^{K}\sum_{\substack{k'=1\\k\neq k'}}^{K}\hat{p}_{jk}\hat{p}_{jk'} = \sum_{k=1}^{K}\hat{p}_{jk}(1-\hat{p}_{jk})$$

- Cross-entropy

$$-\sum_{k=1}^{K}\hat{p}_{jk}\log\hat{p}_{jk}$$



Gini index and cross-entropy are more sensitive to change in $\hat{p}_{mk}$

Example: Email spam

- Information from 4601 email messages, 57 features
- Predict whether the email was “spam”
- Learn rules such as

$$\text{if } (\texttt{\%george} < 0.6) \;\&\; (\texttt{\%you} > 1.5) \quad \text{then } \texttt{spam} \text{ else } \texttt{email}.$$

$$\text{if } (0.2 \cdot \texttt{\%you} - 0.3 \cdot \texttt{\%george}) > 0 \quad \text{then } \texttt{spam} \text{ else } \texttt{email}.$$

- Using CART for classification tree

TABLE 1.1. *Average percentage of words or characters in an email message equal to the indicated word or character. We have chosen the words and characters showing the largest difference between* `spam` *and* `email`.

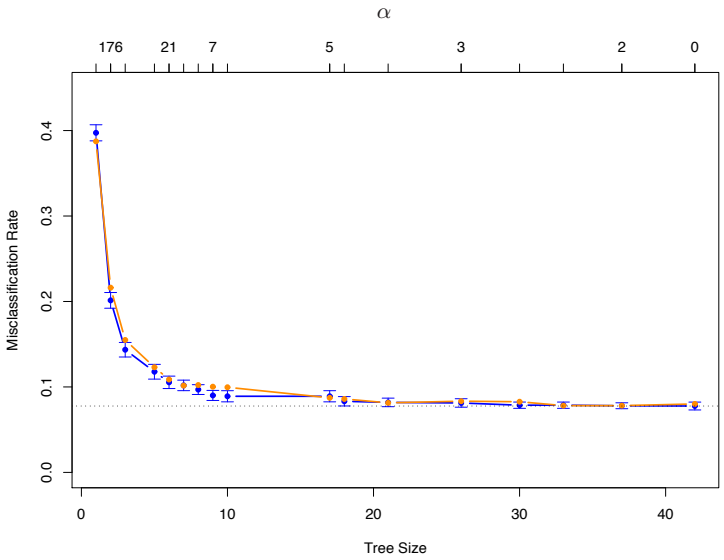
	george	you	your	hp	free	hpl	!	our	re	edu	remove
spam	0.00	2.26	1.38	0.02	0.52	0.01	0.51	0.51	0.13	0.01	0.28
email	1.27	1.27	0.44	0.90	0.07	0.43	0.11	0.18	0.42	0.29	0.01

Detailed description: https://archive.ics.uci.edu/ml/datasets/spambase

Example: Email spam

TABLE 9.3. *Spam data: confusion rates for the 17-node tree (chosen by cross–validation) on the test data. Overall error rate is 9.3%.*

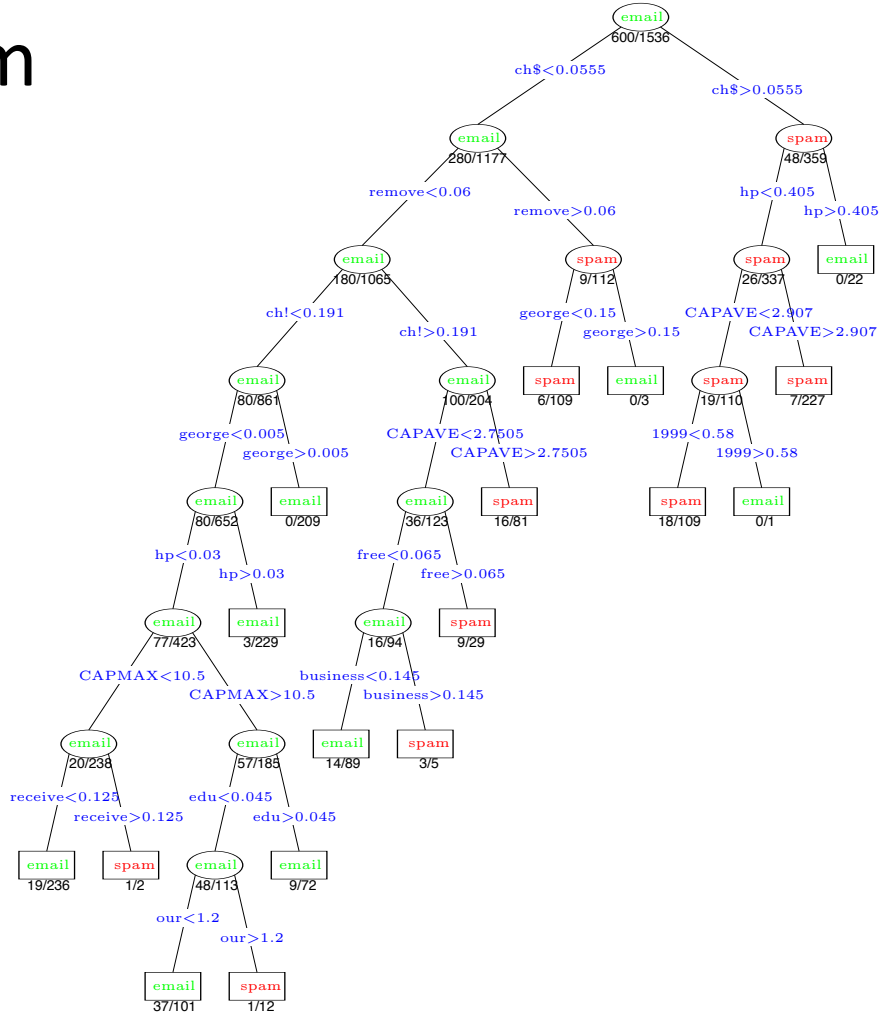
	Predicted	
True	email	spam
email	57.3%	4.0%
spam	5.3%	33.4%



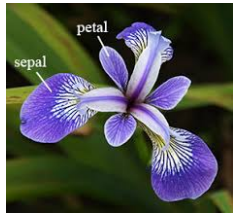
Cross-validation choose α that gives a tree size 17

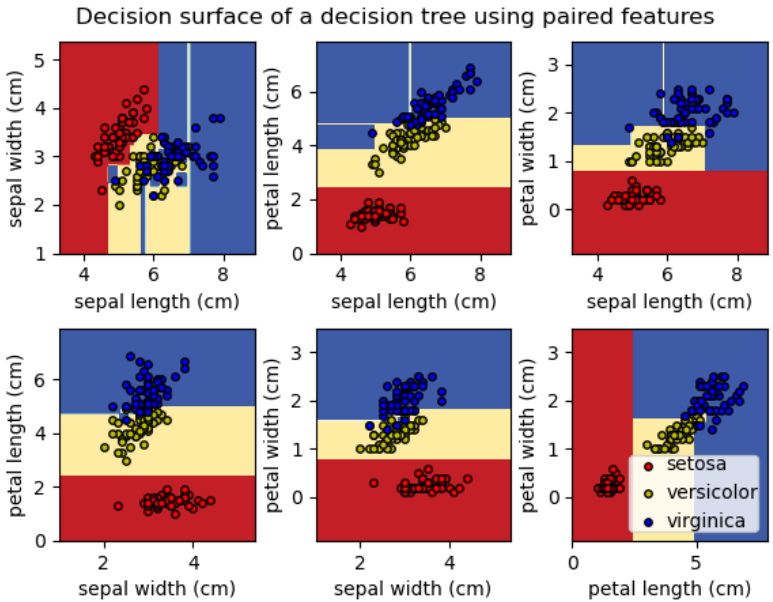
Reference: Chapter 9.2.5 in "Elements of Statistical Learning".

Example: Email spam



Demo: CART for Fisher's iris data

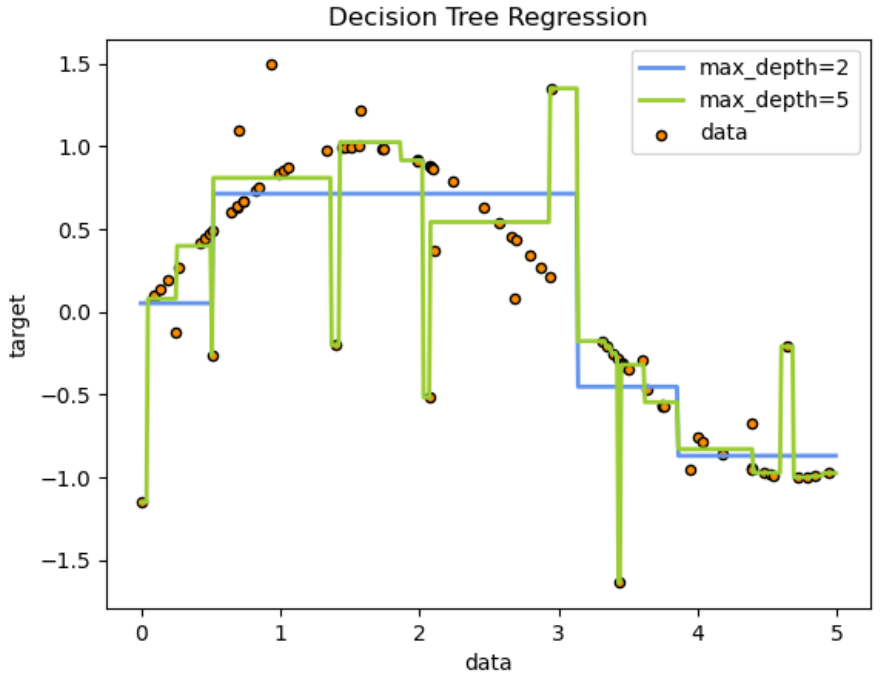




https://scikit-learn.org/stable/auto_examples/tree/plot_iris_dtc.html

Demo: Decision tree regression

- Use decision tree to fit a sine curve
- When the maximum depth of the tree is set too high, the decision trees learn too fine details of the training data and learn from the noise, i.e. overfit.

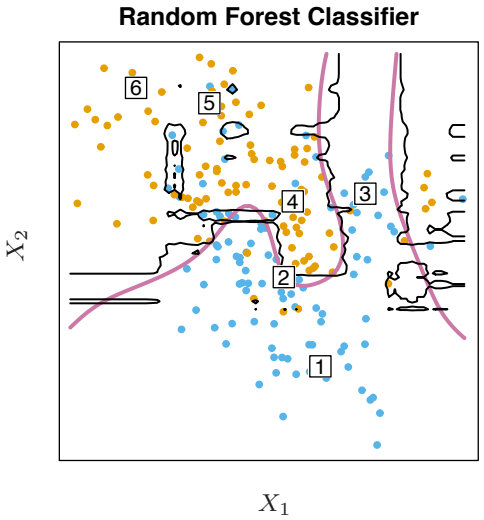


https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html#sphx-glr-auto-examples-tree-plot-tree-regression-py

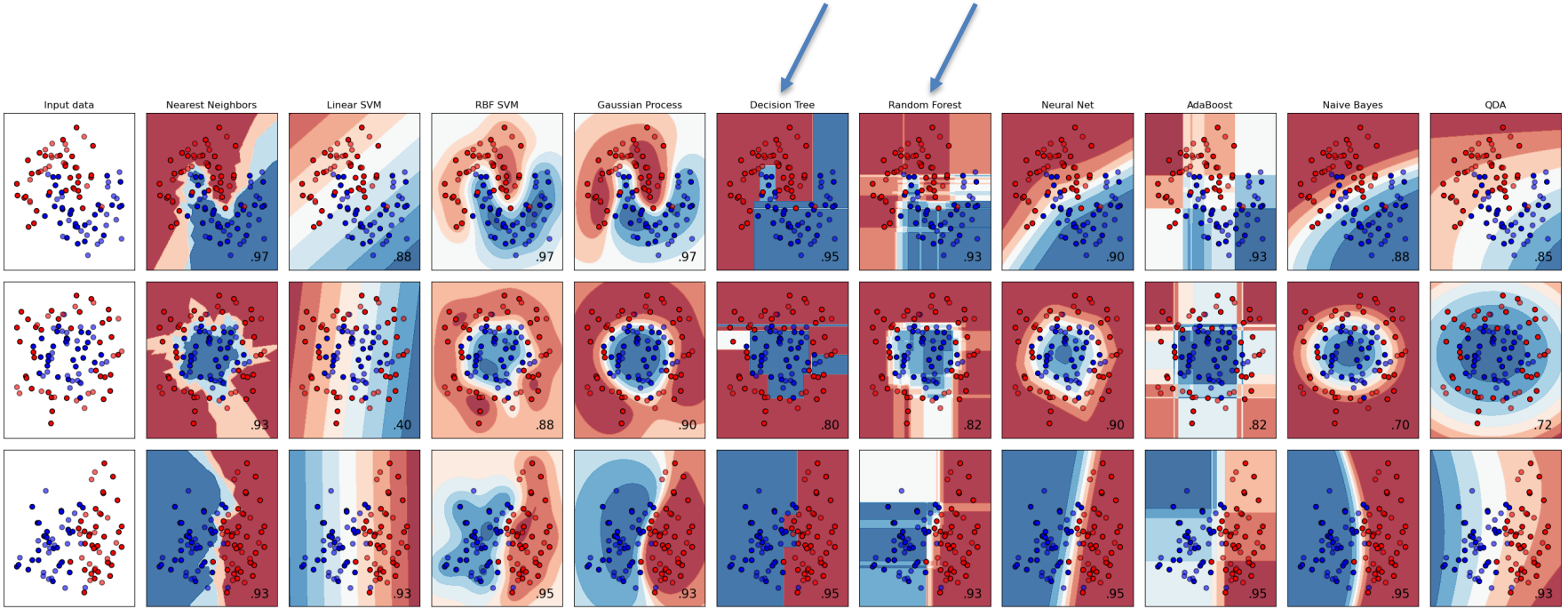
Pros and Cons of decision tree

- Trees (if grown large enough) can capture complex structures
- A main issue of the tree-based method is noisy outcome
- They benefit from averaging

- **Random forest** for regression for classification
  - **Bootstrap** samples from training data
  - Grow a tree for each batch of bootstrap samples
    - Regression: Average trees' predictions
    - Classification: Take majority vote across trees



Classifier comparison

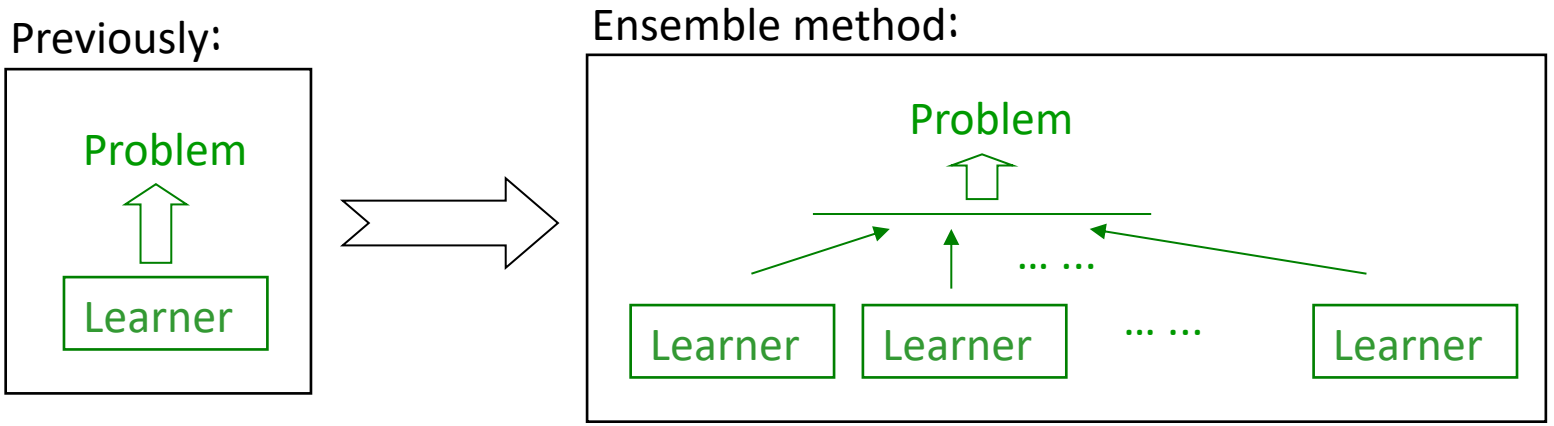


Recall from last lecture: Combination of methods

- There is no algorithm that is always the most accurate

"Can a set of weak learners create a single strong learner?"

- Kearns and Valiant (1988, 1989)



- Different learners use different
  - Algorithms, Hyperparameters, Representations, Training sets, Subproblems

Two main approaches

- Boosting
  - Run weak learner on **weighted** example set
  - Combine weak hypotheses linearly
  - Require knowledge on the performance of weak learner

- Bagging
  - Run weak learners on bootstrap replicates of the training set
  - Average weak leaners
  - Reduces variance

Bootstrap

- Idea: in statistics, we learn about the characteristics of the population by taking samples.

- As the sample represents the population, analogous characteristics of the sample should give us information about the population characteristics.

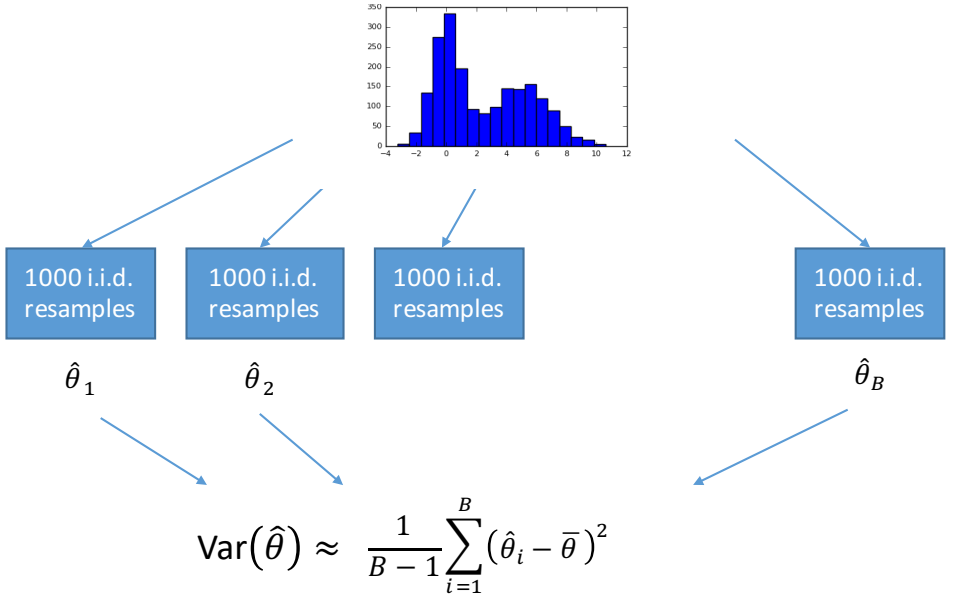
- Bootstrapping learns about the sample characteristics by taking **resamples** and use the information to infer the population

- Resample: we retake samples **with replacement** from the original samples.

- Provides a powerful too to calculate the standard error of an estimator, construct confidence intervals, and many other uses.

Example: Estimate variance of estimator

- We would like to decide the variance of an estimator $\hat{\theta}$
- Use data to generate B batches of new samples and evaluate $\hat{\theta}$ for each resampled batch



Random forest

For $b = 1, \dots, B$

Draw bootstrap sample Z_b of size N from training data

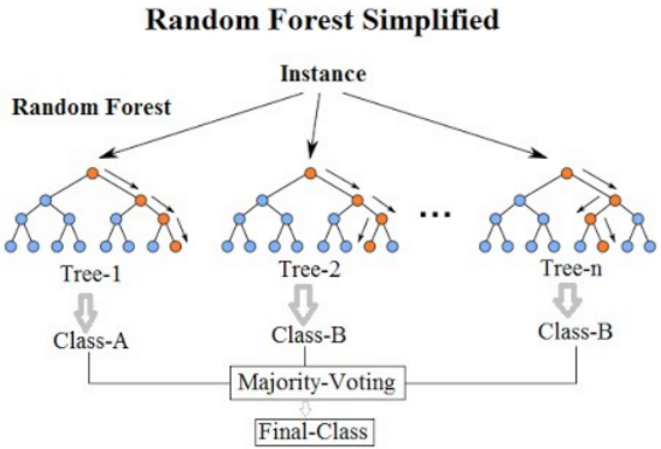
Grow a random forest tree T_b for the boostraped data by

- Select v variables at random from p variables
- Pick the best variable/split among the v
- Split the node into two children nodes

Output ensemble of trees $T_b, b = 1, \dots B$

To make prediction for a new point x

- Regression: $\frac{1}{B}\sum_{b=1}^{B} T_b(x)$
- Classification: majority vote of $\{T_b(x)\}, b = 1, \dots, B$

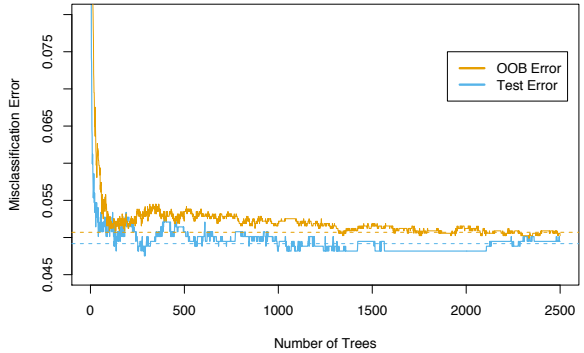


Considerations in random forests

- Effect of averaging
  - Average of B i.i.d. random variables, each with variance σ^2, has variance σ^2/B.
  - If the variables are dependent with positive correlation ρ, the variance of the average is $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$
- The amount of correlation between pairs of bagged tree limit the benefit of averaging
- Before each split, select $\nu \leq p$ of the input variables at random as candidates of splitting.
  - For classification, $\nu \approx \lfloor\sqrt{p}\rfloor$
  - For regression, the default value $\nu \approx \lfloor p/3 \rfloor$

OOB error

Spam data



- Out-of-bag (OOB) samples:

For each observation $z^i = (x^i, y^i)$, construct its random forest predictor by averaging only those trees corresponding to bootstrap samples in which z^i did not appear.

- OOB error estimate is almost identical to that obtained by N-fold cross validation.

- Help to decide the number of trees: Once OOB error stabilizes, the training can be terminated.

Boosting trees

A tree can be represented as (parameter $\Theta = \{R_j, \gamma_j\}_{j=1}^{J}$)

$$T(x, \Theta) = \sum_{j=1}^{J} \gamma_j \mathbb{I}(x \in R_j)$$

Minimize the cost function

$$\widehat{\Theta} = \text{argmin}_{\Theta} \sum_{j=1}^{J} \sum_{x_i \in R_j} L(y^i, \gamma_j)$$

Forward stagewise optimization for boosting trees

- Solve a sequence of optimization problems

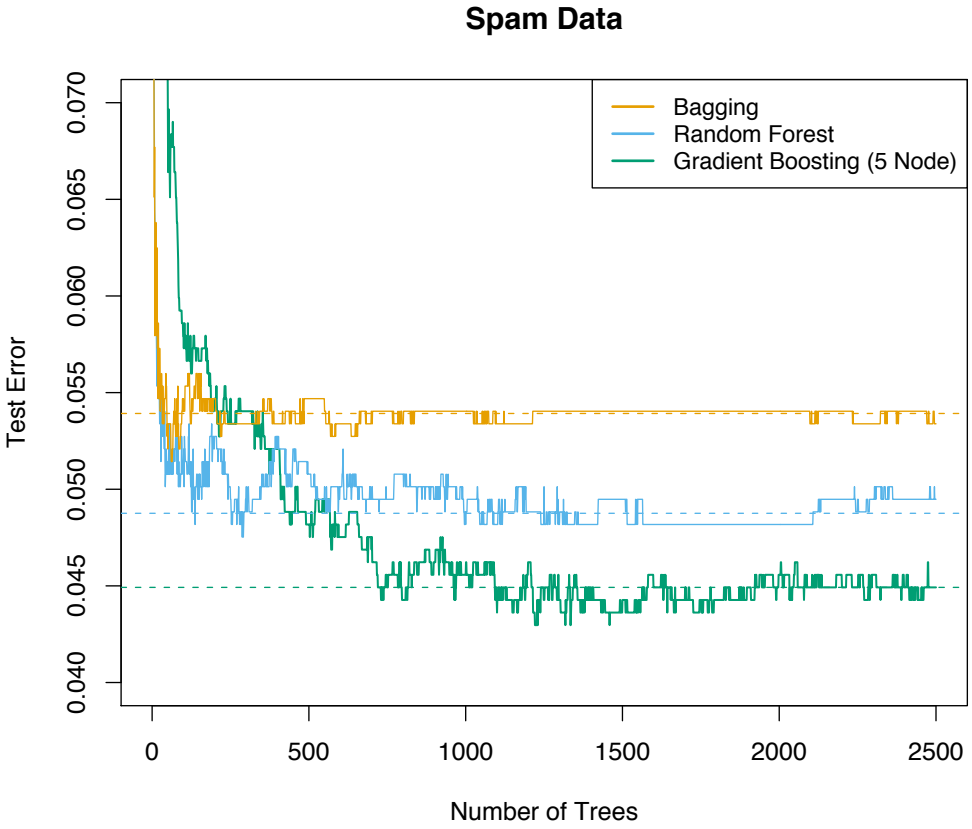
$$\widehat{\Theta}_t = \arg\min_{\Theta_t} \sum_{i=1}^{m} L(y^i, f_{t-1}(x^i) + T(x^i; \Theta_t))$$

- Alternating minimization: finding $\gamma(j)$ given R_j, and then find R_j

- For instance, if we have exponential loss like adaboost

$$\hat{\gamma}_t(j) = \frac{1}{2}\log\frac{\sum_{x^i \in R_{jt}} D_t(i) I(y^i = 1)}{\sum_{x^i \in R_{jt}} D_t(i) I(y^i = -1)}$$

Email spam example (earlier)



California housing price

- Data set (Pace and Barry, 1997) is available from CMU StatLib
- Aggregated data from each of 20,460 neighborhood (1990 census data) in California
- 8 predictors
  - Median income
  - House density
  - Average occupancy
  - Location
  - Average number of rooms
  - Average number of bedrooms

Reference: Chapter 10.14.1 in "Elements of Statistical Learning".

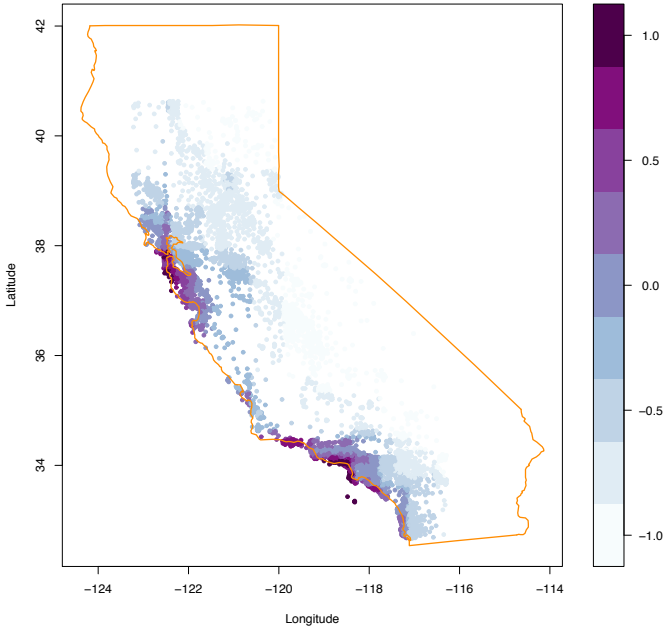


FIGURE 10.17. *Partial dependence of median house value on location in California. One unit is* $100,000, *at* 1990 *prices, and the values plotted are relative to the overall median of* $180,000.

California housing price (cont.)

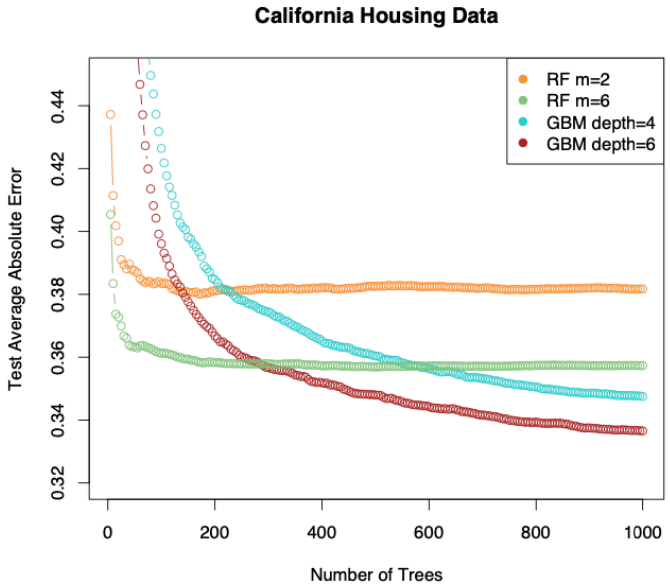


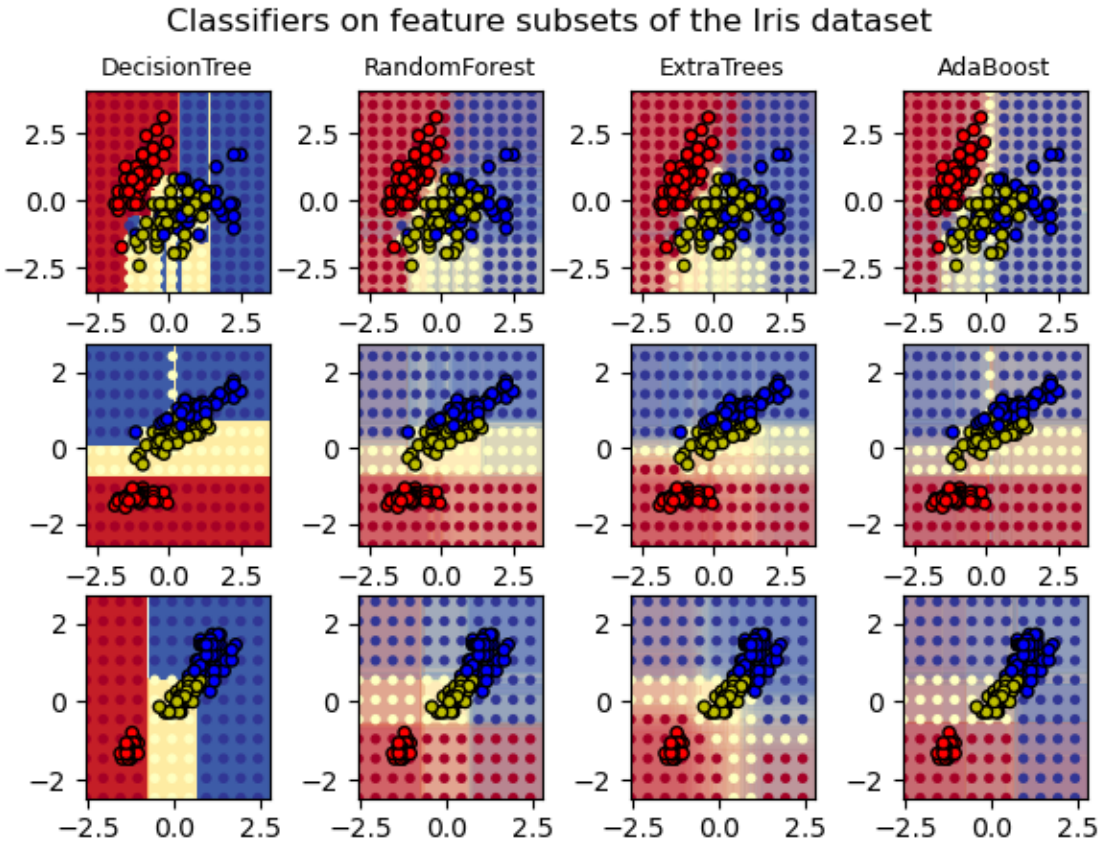
FIGURE 15.3. *Random forests compared to gradient boosting on the California housing data. The curves represent mean absolute error on the test data as a function of the number of trees in the models. Two random forests are shown, with* $m = 2$ *and* $m = 6$*. The two gradient boosted models use a shrinkage parameter* $\nu = 0.05$ *in (10.41), and have interaction depths of* 4 *and* 6*. The boosted models outperform random forests.*

- Random forest stabilizes at about 200 trees
- Boosting outperforms random forest when the number of trees further grows

Demo: Fisher's iris data

- In the first row, the classifiers are built using the sepal width and the sepal length features only, on the second row using the petal length and sepal length only, and on the third row using the petal width and the petal length only.
- Extra tree: use whole sample, average using multiple trees
- Random forest: bootstrap

https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_iris.html#sphx-glr-auto-examples-ensemble-plot-forest-iris-py




SUMMARY