Bagging

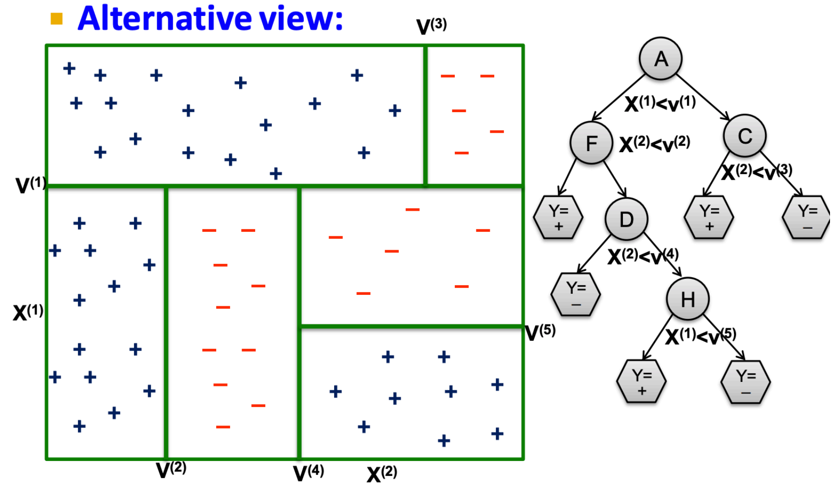
CSC 461: Machine Learning

Fall 2021

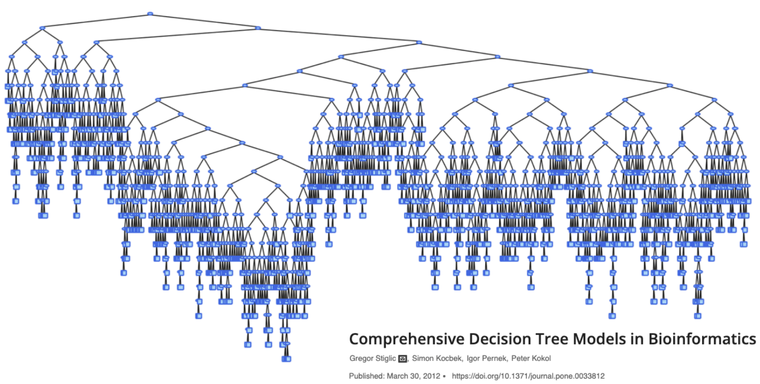
Prof. Marco Alvarez
University of Rhode Island

Feature space

- **Alternative view:**



Jure Leskovec, Stanford CS246: Mining Massive Datasets, http://cs246.stanford.edu



Complicated decision boundaries ==> Overfitting

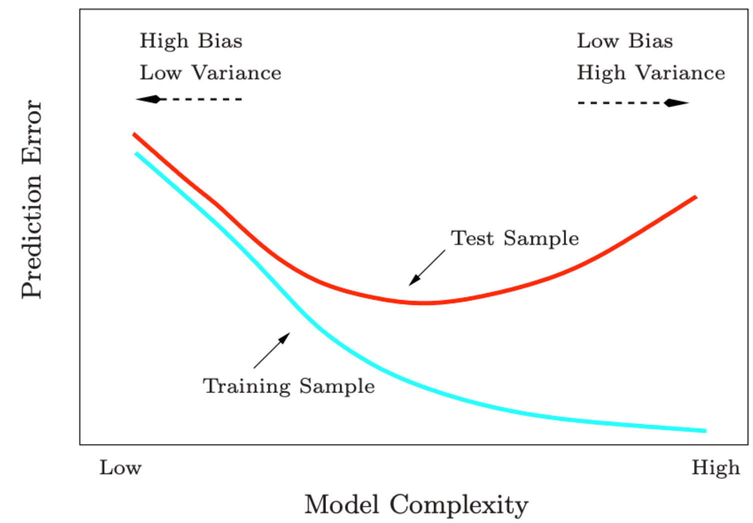
Trees problems

- Overfitting
- Unstable
  - ✓ slight changes of the data => different tree structures

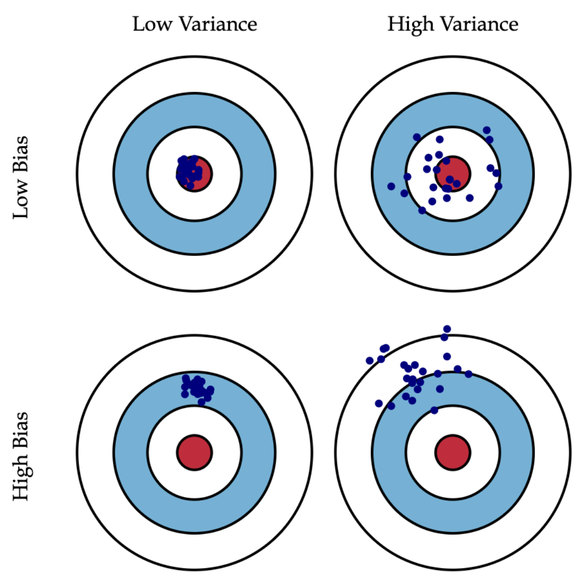
Bias-Variance decomposition

- Expected loss
  - ✓ **bias**: how wrong the expected prediction is
  - ✓ **variance**: the amount of variability in the predictions
  - ✓ **Bayes error**: the inherent unpredictability of the targets (e.g. noise)

$$\mathbb{E}[(y-t)^2] = \underbrace{(y^* - \mathbb{E}[y])^2}_{\text{bias}} + \underbrace{\mathrm{Var}(y)}_{\text{variance}} + \underbrace{\mathrm{Var}(t)}_{\text{Bayes error}}$$



The Elements of Statistical Learning, Hastie, Tibshirani, Friedman, 2nd Ed.



http://scott.fortmann-roe.com/docs/BiasVariance.html

Weak learners

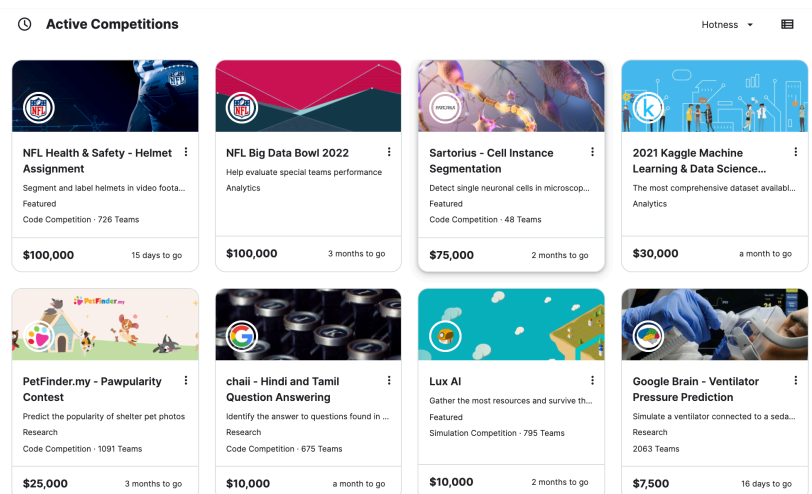
- Weak learners generally have low variance and don't overfit (e.g. shallow trees)
  - ✓ however, they show high bias and are subject to underfitting
- Idea:
  - ✓ combine models to reduce variance

Ensembles

- Set of hypotheses (e.g. classifiers)
  - ✓ individual predictions are combined into a final prediction, e.g. majority vote
- **Bagging (bootstrap aggregation)**
  - ✓ train models independently (**in parallel**) on random subsets of data
  - ✓ variance-reduction technique
- **Boosting**
  - ✓ train **weak** models **sequentially**, each focusing on examples misclassified by previous models
  - ✓ bias-reduction technique

Netflix prize

Kaggle competitions



Bootstrapping

- Assuming a dataset $\mathscr{D}$ with n examples
- Generate m datasets
  - ✓ sample n instances from $\mathscr{D}$ **with replacement** (bootstrap samples)
  - ✓ some elements will appear multiple times
  - ✓ some elements may not appear at all

probability of each element not being selected: $\left(1 - \frac{1}{n}\right)^n$

36.8% for large n

Exercise

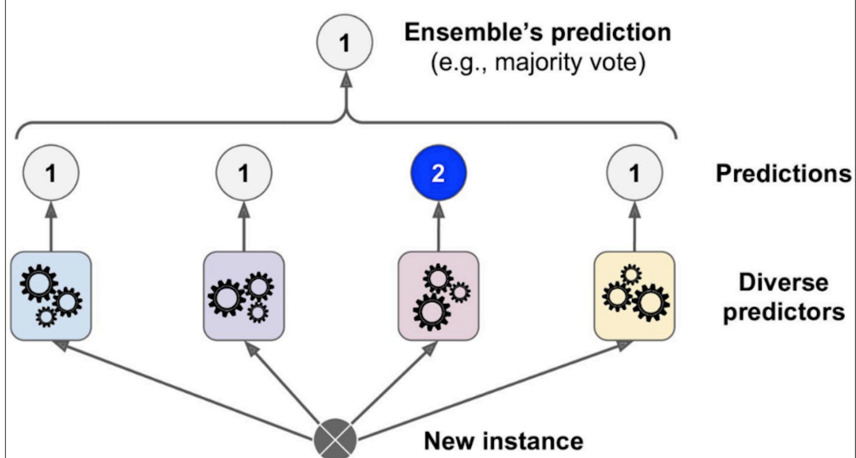
- Write a script that generates a random sequence of N elements and creates M bootstrap samples from that sequence
  - ✓ can use `random.randint` and `random.choices`

Random Forests

Bootstrapping

https://www.bpesquet.fr/mlhandbook/algorithms/decision_trees_and_random_forests.html

Inference



https://www.bpesquet.fr/mlhandbook/algorithms/decision_trees_and_random_forests.html

Random Forest

- Ensemble
  - ✓ create m trees trained from bootstrap "samples"
  - ✓ majority vote for prediction
- Benefits
  - ✓ reduces overfitting — low variance, however it has little effect on bias
- Combines **example diversity** with **feature diversity**

Algorithm

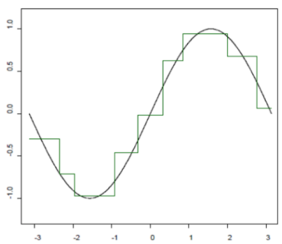
Algorithm RandomForest(D, T, d) – train an ensemble of tree models from bootstrap samples and random subspaces.

Input : data set D; ensemble size T; subspace dimension d.
Output : ensemble of tree models whose predictions are to be combined by voting or averaging.

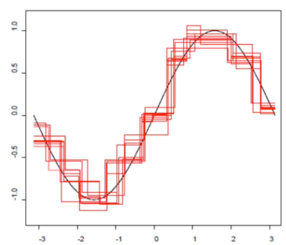
for $t = 1$ to T **do**
  build a bootstrap sample D_t from D by sampling $|D|$ data points with replacement;
  select d features at random and reduce dimensionality of D_t accordingly;
  train a tree model M_t on D_t without pruning;
end
return $\{M_t | 1 \le t \le T\}$

from: Machine Learning Making Sense of Data, http://people.cs.bris.ac.uk/~flach/mlbook/

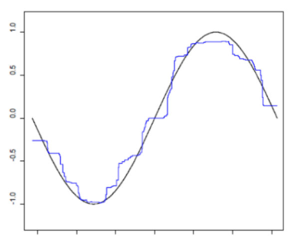
Regression example



1 tree

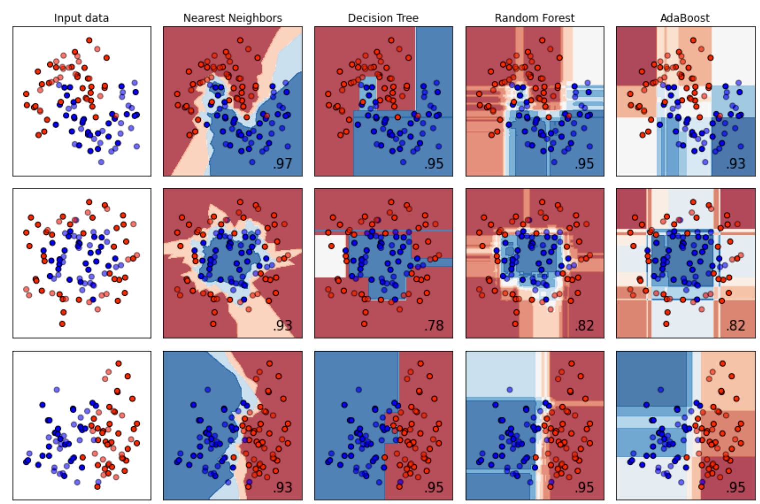


10 trees



average

Comparing classifiers



Issues

- Fitting ensembles can be computationally intensive
  - ✓ can use *max_depth* to alleviate
- Naively averaging or taking a majority both may not be optimal
  - ✓ stay tuned: **boosting**