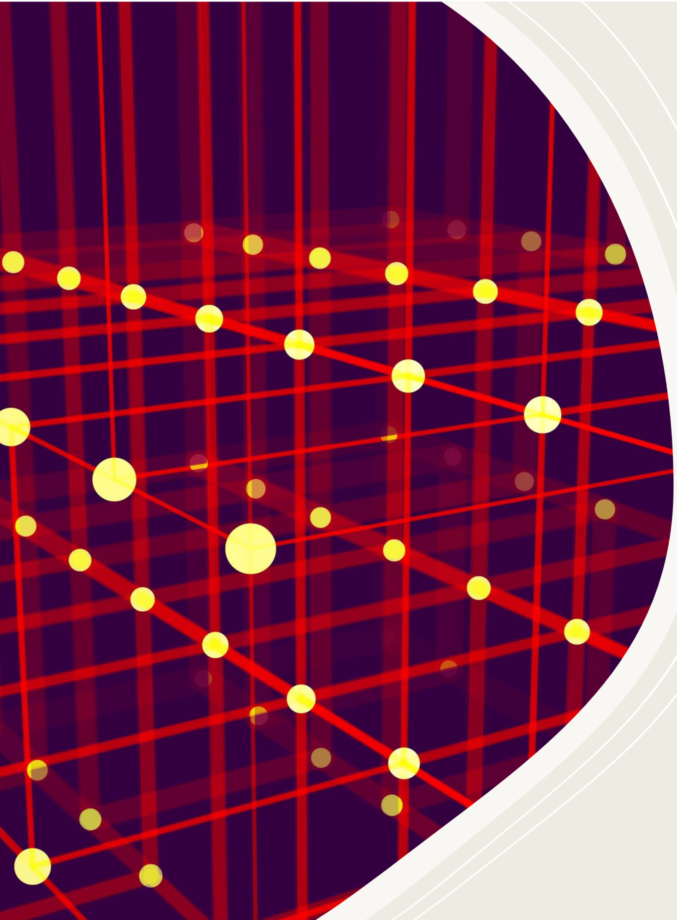


CPSC 340 – Tutorial 3

Slides courtesy of Nam Hee Kim

University of British Columbia

Agenda

- Ensemble Methods
  - Bootstrap Aggregating (*Bagging)*
  - Boosting
  - Averaging
  - Random Forest
- Unsupervised learning: Clustering
  - KMeans
  - DBSCAN

Ensemble methods

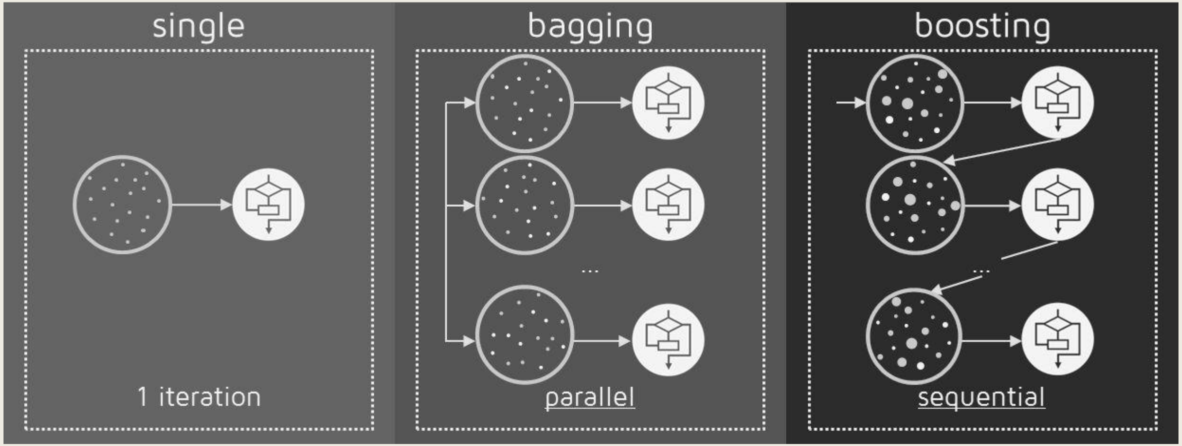
- Idea: use classifiers as building blocks for more complex model
- Goal: reduce bias/variance by combining several base models

Reminder: bootstrapping = random sampling with replacement

Ensemble methods: bagging, boosting and averaging

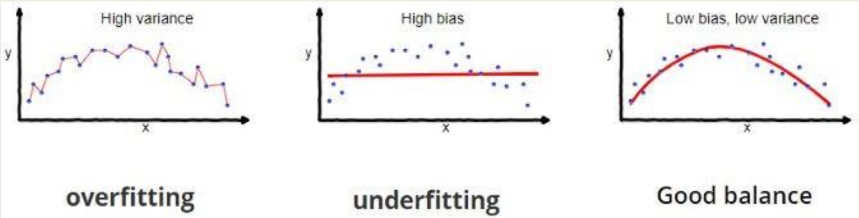
- **B**ootstrap **Agg**regat**ing** (bagging): base models are trained **independently** on bootstrap samples, and then combined through majority voting or averaging
- **Boosting**: base models are trained **sequentially**, and then combined through majority voting or averaging. In boosting, we perform "weighted" resampling, so observations that were misclassified by the previously trained base model have a greater chance of being sampled
- **Averaging**: base models are trained **independently**, the final predictions are determined through majority voting or averaging. **Stacking** is a variation of this method in which base models' predictions serve as features for a *meta-classifier*

Ensemble methods: bagging vs. boosting



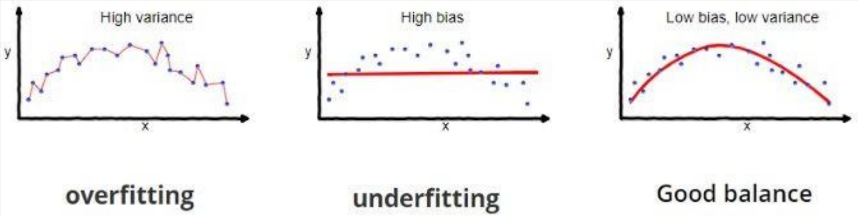
Ensemble methods: bagging vs. boosting
Quick Questions

1. What do we aim to reduce with bagging, bias or variance?
2. How about in boosting?



Ensemble methods: bagging vs. boosting Quick Questions

1. What do we aim to reduce with bagging, bias or variance?
    a. reduce variance, which helps controlling overfitting
2. How about in boosting?
    a. both, boosting tries to improve the training error for classifiers with high E_{train}



Random Forest

Idea: bagging using random trees as base classifiers

Random trees: For each tree, randomly sample a small number of features to be considered as candidates split variables.

Given *n* (# of training examples), *d* (# of features), *m* (depth), and *t* (# of test examples):

a) What is the cost of fitting a random tree that uses $\sqrt{d}$ random features?
b) What is the cost of predicting using a random tree?

Random Forest

Training:

- Our normal cost for fitting a decision tree is $O(mnd \log(n))$, and the factor of d comes from searching through all features for each split. If we only search $\sqrt{d}$ features this will be reduced to $O(mn\sqrt{d} \log(n))$.

Predicting:

- For each training example t, we only do an $O(1)$ operation at each level of the decision tree to decide whether we satisfy the rule. This gives a total cost of $O(tm)$.

Unsupervised Learning

- Only have x_i values, but no explicit target labels.
- **Clustering:** trying to discover the underlying structures (clusters) in the data, such as grouping customers by purchasing behavior.
- **Association:** trying to identify rules that describe large portions of the data, such as people that buy X also tend to buy Y.

K-Means

Idea: group points based on how similar they are to the "mean" of the cluster Hyperparameter: *k* number of clusters

Fit:

1. Initialize *k* clusters by creating *k* centroids (usually randomly)
2. Assign each data point to closest mean
3. Update the means bases on the assignment
4. Repeat until convergence (for example, when points don't change clusters)

Predict:

1. Assign test example to the nearest mean

K-Means: Quick questions

- What typically happens with E_{train} as k increases?
- What is the space complexity for K-Means?

K-Means: Quick questions

What typically happens with E_{train} as k increases?

- E_{train} should decrease

What is the space complexity for K-Means?

- O(kd). You need to store k cluster centres, each is a vector of length d

K-Means: Quick questions

What typically happens with E_{train} as k increases?

- E_{train} should decrease

What is the space complexity for K-Means?

- O(kd). You need to store k cluster centres, each is a vector of length d

K-Means: Practice question

Suppose we fit an unsupervised k-means model to a dataset and obtained the following means:

$$W = \begin{bmatrix} 8.0 & 9.5 \\ 1.0 & 7.0 \\ 13.5 & 3.0 \end{bmatrix}$$

Which cluster would a test example x = [9 4] get assigned to?

K-Means: Practice question

Suppose we fit an unsupervised k-means model to a dataset and obtained the following means:

$$W = \begin{bmatrix} 8.0 & 9.5 \\ 1.0 & 7.0 \\ 13.5 & 3.0 \end{bmatrix}$$

Which cluster would a test example x = [9 4] get assigned to?

(8-9)^2 + (9.5-4)^2 = 31.25

(1-9)^2 + (7-4)^2 = 73

(13.5-9)^2 + (3-4)^2 = 21.25

Density-based Spatial Clustering (DBSCAN)

Idea: clusters are defined by dense regions (non-dense regions don't get clustered), merge all neighboring core points to form clusters

Hyperparameters: ε - threshold used to determine if another point is a neighbor

minNeighbors - number of neighbors needed to say that a region is dense enough to be a cluster. "Core" points have at least minNeighbors

DBSCAN: Quick questions (T/F)

1. For data points to be in a cluster, they must be in a distance threshold to a core point
2. It has strong assumptions for the distribution of data points in dataspace
3. It is a non-parametric model

DBSCAN: Quick questions (T/F)

1. For data points to be in a cluster, they must be in a distance threshold to a core point
- True
2. It has strong assumptions for the distribution of data points in dataspace
- False, DBSCAN can form a cluster of any arbitrary shape and does not have strong assumptions for the spatial distribution of data points
3. It is a non-parametric model
- True

Bonus Questions

1. When choosing k in K-Means, why not just choose the k that leads to the smallest distance (sum of squared distances within clusters)?
2. You decide to use clustering for outlier detection; that is, to detect instances that are very atypical compared to all the rest. How might you do this with K-Means?
3. You decide to use clustering for outlier detection; that is, to detect instances that are very atypical compared to all the rest. How might you do this with DBSCAN?