Ensemble Classifiers

Bias - how well the model capture the boundary.

Variance - deviation around the boundary. Small perturbations can lead to different decision boundaries.

KNN vs.

Bias is low
Higher variance

Bias - variance tradeoff
Bagging: Bootstrap Aggregation

- Draw $n$ samples $S_1, \ldots, S_n$ from $D$ with replacement.
- For each $S_i$, learn a classifier $M_i$.
- Combine classifiers $M_1, \ldots, M_n$ using majority voting.

Test case $z$:
- $\hat{y} = M(z)$
- $M_i(z) \rightarrow c_i$ for each $i = 1, \ldots, n$.
- Majority vote $c_i$'s.

Variance reducing:
- Can not typically reduce bias.

Averaging effect.
Boosting: can reduce bias

biased sampling

\[ x_1, x_2, \ldots, x_n \]

\[ S_1 \rightarrow M_1 \]

\[ S_2 \rightarrow S_3 \rightarrow \ldots \rightarrow S_{k-1} \rightarrow M_k \]

weights \( \alpha_1, \alpha_2, \ldots, \alpha_k \)

\[ M(z) = \text{weighted vote according to } \alpha \]

models \( M_1, M_2, \ldots, M_k \)

\[ Z \rightarrow M_1 \rightarrow C_1 - \alpha_1 \]

\[ M_2 \rightarrow C_2 \]

\[ M_3 \rightarrow C_1 \]

\[ \vdots \]

\[ M_k \rightarrow C_k \]

\[ Z \text{ sum } \rightarrow \text{ pick higher weight class} \]
AdaBoost: Adaptive Boosting

\( \hat{w}_t \in \mathbb{R}^n \)  
weight for each point \((x_1, x_2, \ldots, x_n)\)

\( z \in \mathbb{R}^k \)  
weight for each classifier \((M_1, M_2, \ldots, M_k)\)

\( t = 0 \)

\( \hat{w} = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}) \)

for \( t = 1, \ldots, k \)

\( s_t \leftarrow \) draw a sample according to \( \hat{w}_t \)

\( y \leftarrow M_t \)

test \( M_t \) on entire training set \( D \)

compute the weighted error rate \( \hat{\varepsilon}_t \)

\[ \hat{\varepsilon}_t = \frac{1}{n} \sum_{i=1}^{n} \hat{w}_i \cdot I(M_t(x_i) \neq y_i) \]

(sum of the weights of misclassified points)

\( \varepsilon_t \approx 0.5 \)

\( z_t \rightarrow 0 \)

\( \varepsilon_t \) is very small \( z_t \) is high.

if \( \varepsilon_t < 0.5 \)

\[ a_t = \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right) \]

better than random
Re-weight the points

$$\omega_i = \frac{\omega_i}{\new{\omega_i}} \exp \left\{ \alpha_i \cdot \frac{1}{\xi_i} \left( \frac{1}{\xi_i} \right) \right\}$$

for a correctly predicted point

$$\frac{1}{\xi_i} \left( \frac{1}{\xi_i} \right) = 0$$

weight remains the same.

for misclassified

$$\omega_i = \omega_i \times \exp \left( \alpha_i \right)$$

$$\omega_i = \frac{\omega_i}{\new{\omega_i}} \left( 1 - \frac{3}{\xi_i} \right)$$

$$\omega_i = \omega_i \left( 1 - \frac{1}{\xi_i} \right)$$

\(\xi_i \geq 0.5\)

bad classifier

expose to make many mistakes

\(\xi_i \leq 0.001\)

very good classifier

since \(X_i\) was misclassified

we want very high

weight for near rare.
\[
\begin{pmatrix}
0.1 & 0.1 & \cdots & 0.1 \\
\overbrace{0.05} & 0.2 & 0.2 & \overbrace{0.05} & 0.01 & \cdots
\end{pmatrix}
\]

**Exam II**

3 exams posted on piazza

1) **LDA**

\[ LDA \rightarrow S \quad \overbrace{S'B} \quad \tilde{S}'(\mu, \Sigma) \]

2) **SVM**

\[ \text{A value given} \]

\[ \text{find } h : (u, b) \]

\[ \text{what is the margin?} \]

\[ \text{test point } \rightarrow \text{classify} \]

\[ \text{Kernel} \]

\[ \omega \cdot x \]

\[ b \cdot v \]

3) **Bayes classifier** ➔ **FHM**

\[ \text{Naive} \]

\[ P(x | c_i), \quad P(x | c_2) \]

\[ P(c_1 | x) = \frac{P(x | c_1) \cdot P(c_1)}{P(x)} \]

\[ \mu_1, \Sigma_1, \quad \mu_2, \Sigma_2, \quad \mu_3, \Sigma_3 \]

Use multivariate normal.
1) Decision Trees
   Evaluate different split points
   \[
   A \leq \alpha
   \]
   \( \rightarrow \) Europe, Asia, etc...

5) KNN
   Find \( K \) neighbors
   Majority class
   Distance metric

6) Evaluation
   Error rate
   F-measure
   ROC
   k-fold cross validation / bootstrapping
   \( t \)-test
   Bagging / Boosting
   \( \omega \) given
   Distribution \( \omega \) random
   \( (\omega, 0.1, 0.2, 0.5, 0.7, 0.1) \)
   \( E_t \) and \( a_t \)
   \[ \omega \rightarrow X_n \]
   \( 0.9 \rightarrow X_n \)
   \( 0.5 \rightarrow \)
Clustering: Unsupervised learning

- No classes.

- Natural groups??
  - Within a group, points are similar.
  - Across groups, points should be dissimilar.

- How many groups?

- \( d \rightarrow \text{high} \Rightarrow \mathbb{R}^d \)

- Shape-based/density-based cluster

- Scarsity effect / find lower dimensional projections with good clusters

- Subspace clustering
1) **Partitioned/Representative-based clustering**

\[ \begin{align*}
D & \quad k \\
\text{dataset} & \quad \# \text{of clusters} \\
C_1, C_2, \ldots, C_k & \quad \text{Partitioning/no overlap} \\
C_i \cap C_j & = \emptyset \\
\bigcup_{i=1}^{k} C_i & = D
\end{align*} \]

\[ \text{Every point must belong to a cluster.} \]

\[ \mathcal{C} = \{ C_1, C_2, \ldots, C_k \} \]

\[ \text{Clustering} \]

\[ J(\mathcal{C}) = f(C_1, C_2, \ldots, C_k) \]

1) **Representative:** \[ C_i \to \mu_i \]

\[ \text{the mean point} \]

\[ \text{represents the cluster} \]

\[ \begin{align*}
\min_{\mathcal{C}} & \quad \text{SSe}(\mathcal{C}) = \sum_{i=1}^{k} \left( \sum_{x_j \in C_i} ||x_j - \mu_i||^2 \right) \\
\text{Sum of squared errors}
\end{align*} \]
NP-hard even for $k = 2$ !!

**k-means Algorithm**

$k = 2$

1) pick $k$ points or random
   call them $u_1, u_2, \ldots, u_k$

2) assign each point $x_i \in D$
   to the closest mean

$$d(x_j, u_i)$$

3) update the $k$ means

$$u_1, u_2, \ldots, u_k$$

now

$$O(kd)$$

Repeat 2 & 3 until convergence

$$O(t \cdot nd)$$

# iterations to converge.