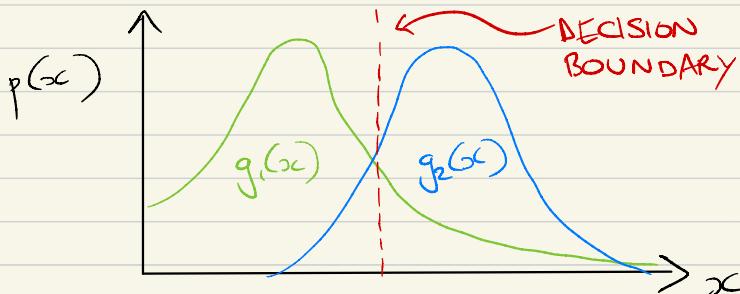


Lecture 19

• Generative vs. Discriminative Classifiers

- * Estimate PDF for each class.
- * Assess probability of new sample belonging to each class.
- * Don't care about all class PDFs.
- * Find a decision boundary that best separates classes.



- ➡ classify new object X
- ① model 1 is better fit than model 2
→ generative.
 - ② X lies on one side of the decision boundary.
→ discriminative.

- Assessing Classification

TYPE I Error \Rightarrow false positive.
 TYPE II Error \Rightarrow false negative.

$$\text{Completeness} \quad = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

(TPR/sensitivity/recall)

$$\text{Contamination} \quad = \frac{\text{FP}}{\text{TP} + \text{FP}}$$

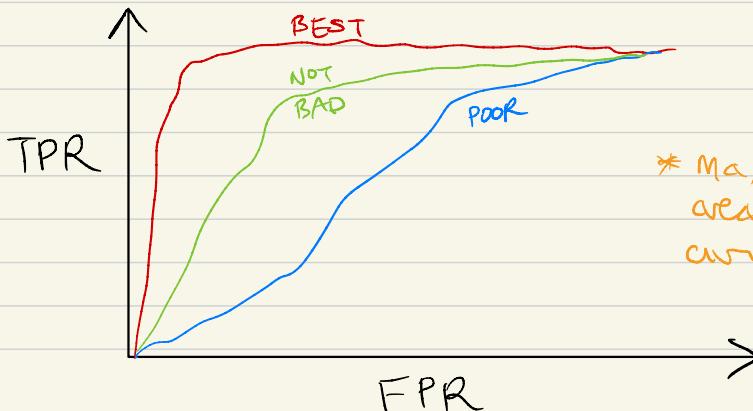
(false discovery rate)

$$\text{Efficiency} = 1 - \text{Contamination}$$

$= \underline{\text{Precision}}$

- Comparing Performance

\Rightarrow we want to maximize the TPR at a given FPR --- ROC curve.



- Gaussian Approximations to Generative Classification

$$p(y_k | x_i) = \frac{p(x_i | y_k) p(y_k)}{\sum_j p(x_i | y_j) p(y_j)}$$

(i) Naive Bayes \leadsto ignore feature covariances
for multi-D data.

$$\hat{y} = \arg \max_{y_k} \frac{\prod_j p(x_i^j | y_k) p(y_k)}{\sum_j \prod_j p(x_i^j | y_j) p(y_j)}$$

(ii) Linear Discriminant Analysis (LDA)

$$g_k(\vec{x}) = \vec{x}^\top \vec{\mu}_k - \frac{1}{2} \vec{\mu}_k^\top \vec{\Sigma}^{-1} \vec{\mu}_k + \ln p(y=y_k)$$

\rightarrow Gaussian Bayes classifier i.e all classes modeled with multi-variate Gaussian PDFs.
All class covariances are EQUAL.

\hookrightarrow gives fast linear dependence of class posterior on data.

(iii) Quadratic Discriminant Analysis (QDA)

$$g_k(\vec{x}) = -\frac{1}{2} \ln \det(\vec{\Sigma}_k) - \frac{1}{2} (\vec{x} - \vec{\mu}_k)^\top \vec{\Sigma}_k^{-1} (\vec{x} - \vec{\mu}_k) + \ln p(y=y_k)$$

\rightarrow relax the requirement that all class covariances are equal.

(iv) GMM Bayes Classifier

\rightarrow relax the assumption that class PDFs are Gaussian --- use a mixture model.

- o K-nearest neighbors Classification

→ Nearest neighbor \Rightarrow if x' is close to x ,
then assign x' to
 x 's known class.

→ K-nearest neighbor \Rightarrow average the assigned
category over the
K-nearest neighbors.