LECTURE 07:
CLASSIFICATION PT. 3

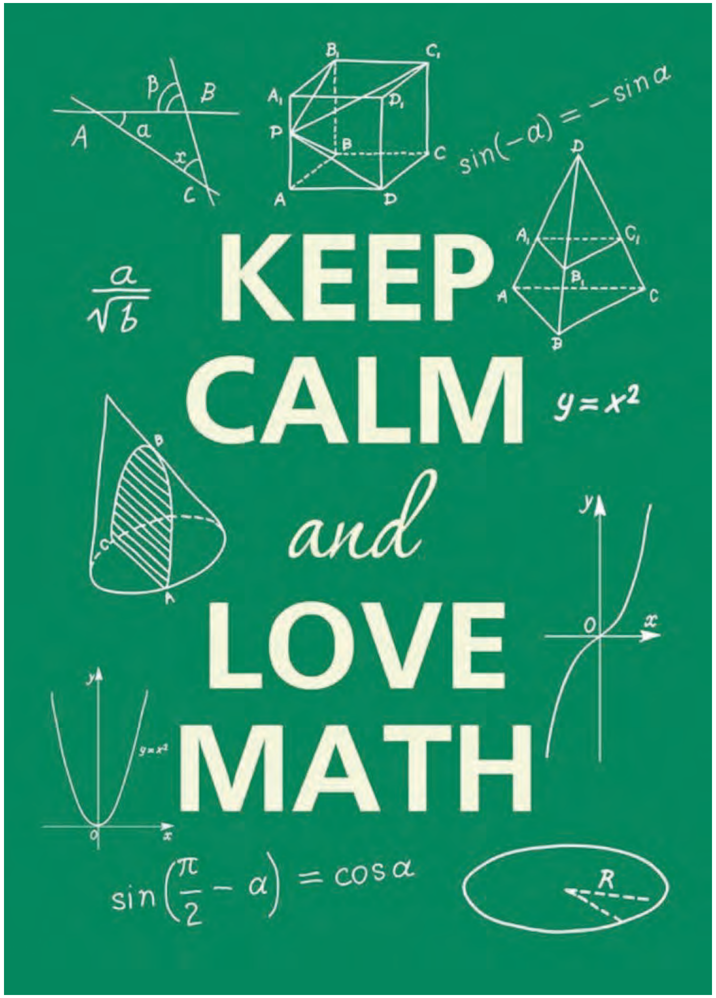
October 02, 2017
SDS 293: Machine Learning

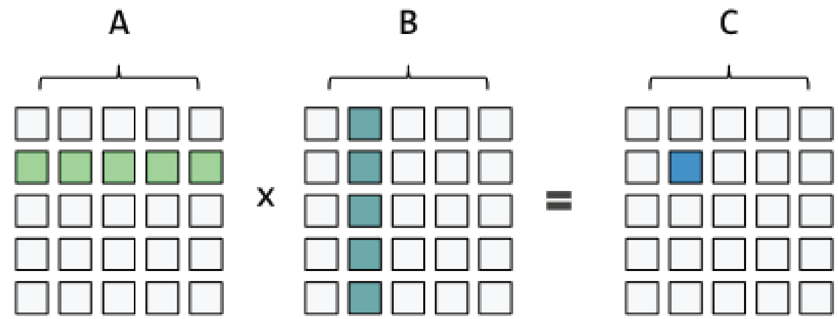
Slides by R.J. Crouser, with a diversion by B.A. Miller

Outline

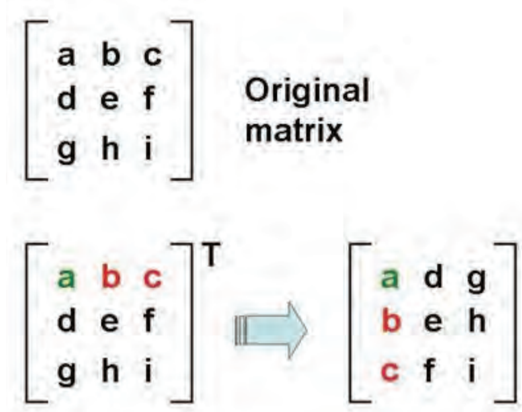
- ✓ Motivation
- ✓ Bayes classifier
- ✓ K-nearest neighbors
- ✓ Logistic regression
  - ✓ Logistic model
  - ✓ Estimating coefficients with maximum likelihood
  - ✓ Multivariate logistic regression
  - ✓ Multiclass logistic regression
  - ✓ Limitations
- Linear discriminant analysis (LDA)
  - Bayes' theorem
  - LDA on one predictor
  - LDA on multiple predictors
- Comparing Classification Methods

Image credit: Rune Anderson

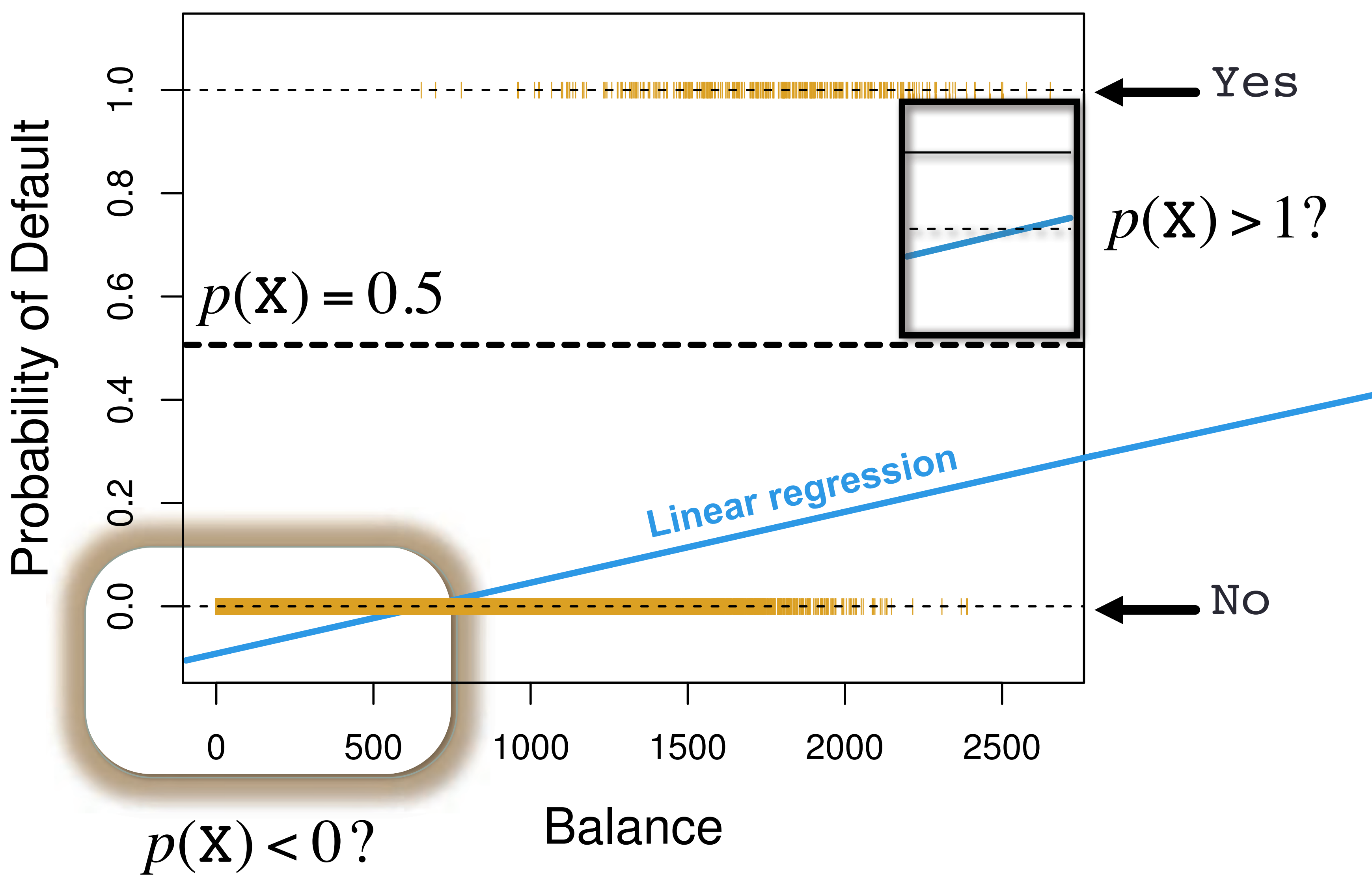




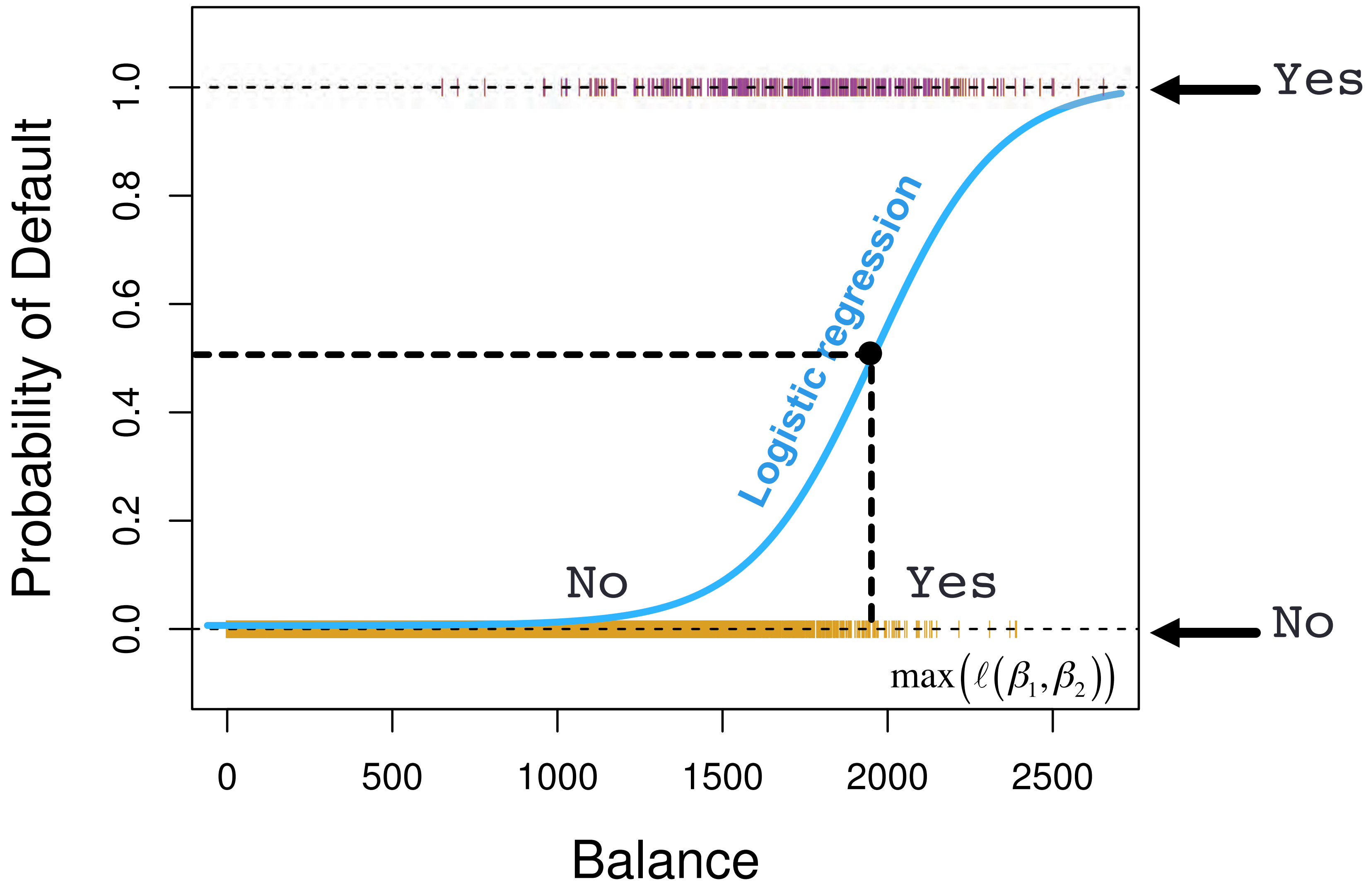
$$A^{-1} = \frac{1}{det\ (A)}\ adj\ (A)$$



Recap: linear regression



Recap: logistic regression



Recap: logistic regression

Question: what were we tying to do using the logistic function?

$$p(\mathrm{X}) = \frac{e^{\beta_0 + \beta_1 \mathrm{X}}}{1 + e^{\beta_0 + \beta_1 \mathrm{X}}}$$

Answer: estimate the *conditional distribution* of the response, given the predictors

Discussion

- What could go wrong?

Flashback: superheroes

According to this sample, we can perfectly predict the 3 classes using only:

Flashback: superheroes

Perfect separation

- If a predictor happens to perfectly align with the response, we call this **perfect separation**

- When this happens, logistic regression will **grossly inflate** the coefficients of that predictor (why?)

$$p(\mathrm{X}) = \frac{e^{\beta_0+\beta_1 \mathrm{X}}}{1+e^{\beta_0+\beta_1 \mathrm{X}}}$$

$$\prod_{i:y_i=1} p(\mathbf{x}_i) \times \prod_{j:yj=1} \left(1-p(\mathbf{x}_j)\right)$$

```
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Another approach

- We could try modeling the distribution of the **predictors** in each class separately:

$$f_k(X) \equiv \Pr(X = x \mid Y = k)$$

"density function of X"

- How would this help?

Refresher: Bayes' rule

$$\Pr(A \mid B) = \frac{\Pr(B \mid A) \times \Pr(A)}{\Pr(B)}$$

Refresher: Bayes' rule

$$\Pr(Y = k \mid X = x) = \frac{\Pr(X = x \mid Y = k) \times \Pr(Y = k)}{\Pr(X = x)}$$

Given that we obs predictor vector x

The probability observe predictor

Given that tl true class is

The probability that the true class is k

The probability that the true class is k

The probability that we observe predictor vector x

Refresher: Bayes' rule

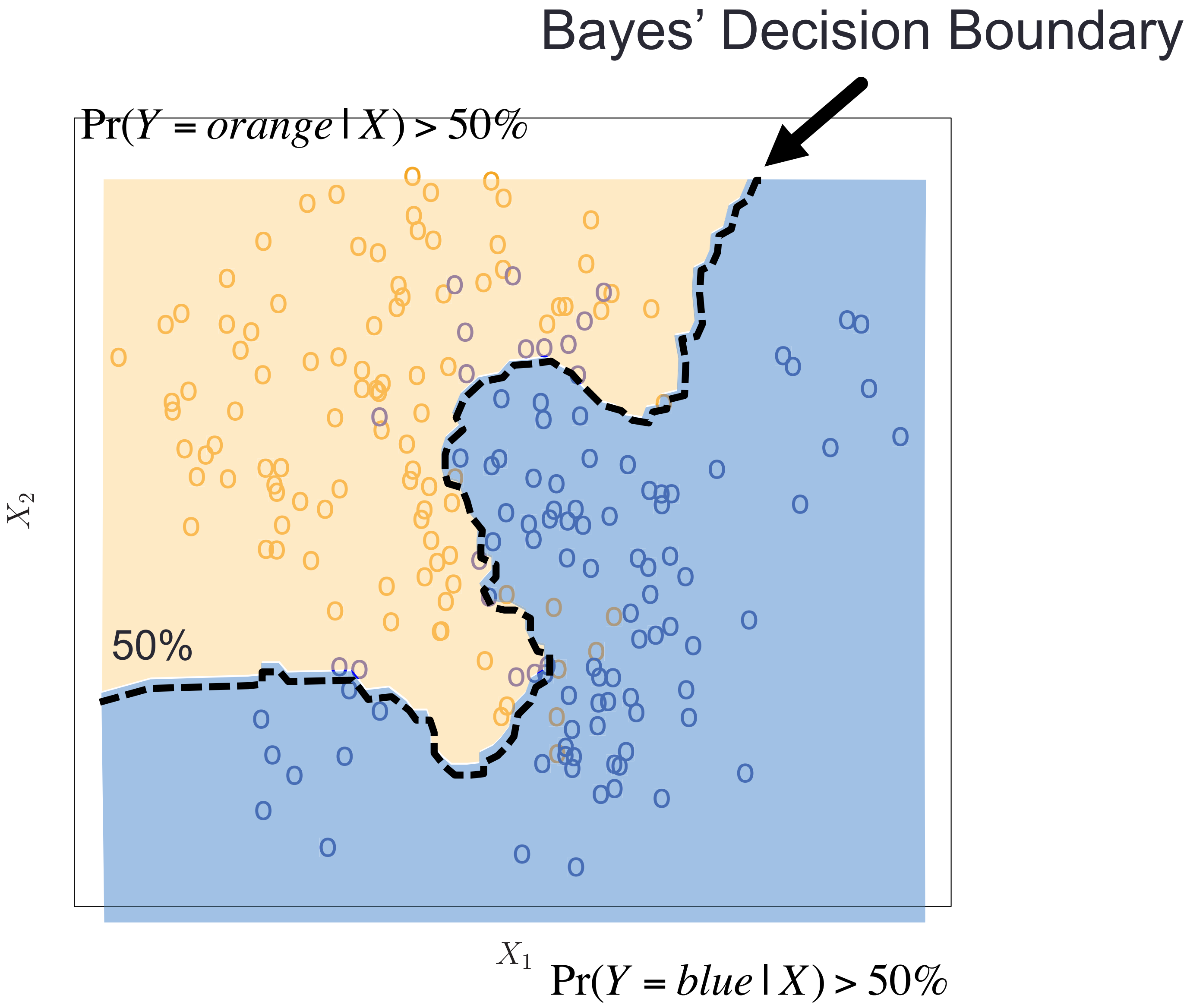
This isn't too hard to estimate*

Just need to estimate the density function!

$$\Pr(Y = k \mid X = x) = \frac{\Pr(X = x \mid Y = k) \times \Pr(Y = k)}{\sout{\Pr(X = x)} \;\; \sum_j \Pr(X = x \mid Y = k) \times \Pr(Y = k)}$$

*Assuming our training data is randomly sampled

Flashback: toy example



Using Bayes' rule for classification

- Let's start by assuming we have just a **single** predictor

- In order to estimate $f_k(X)$, we'll need to make some assumptions about its form

Assumption 1: $f_j(X)$ is normally distributed

- If the density is **normal** (a.k.a. Gaussian), then we can calculate the function as:

$$f_k(x) = \frac{1}{\sqrt{2\pi} * \sigma_k} \times e^{\left(\frac{-1}{2\sigma_k^2} \times (x-\mu_k)^2\right)}$$

where μ_k and σ_k^2 are the mean & variance of the k^{th} class

Assumption 2: classes have equal variance

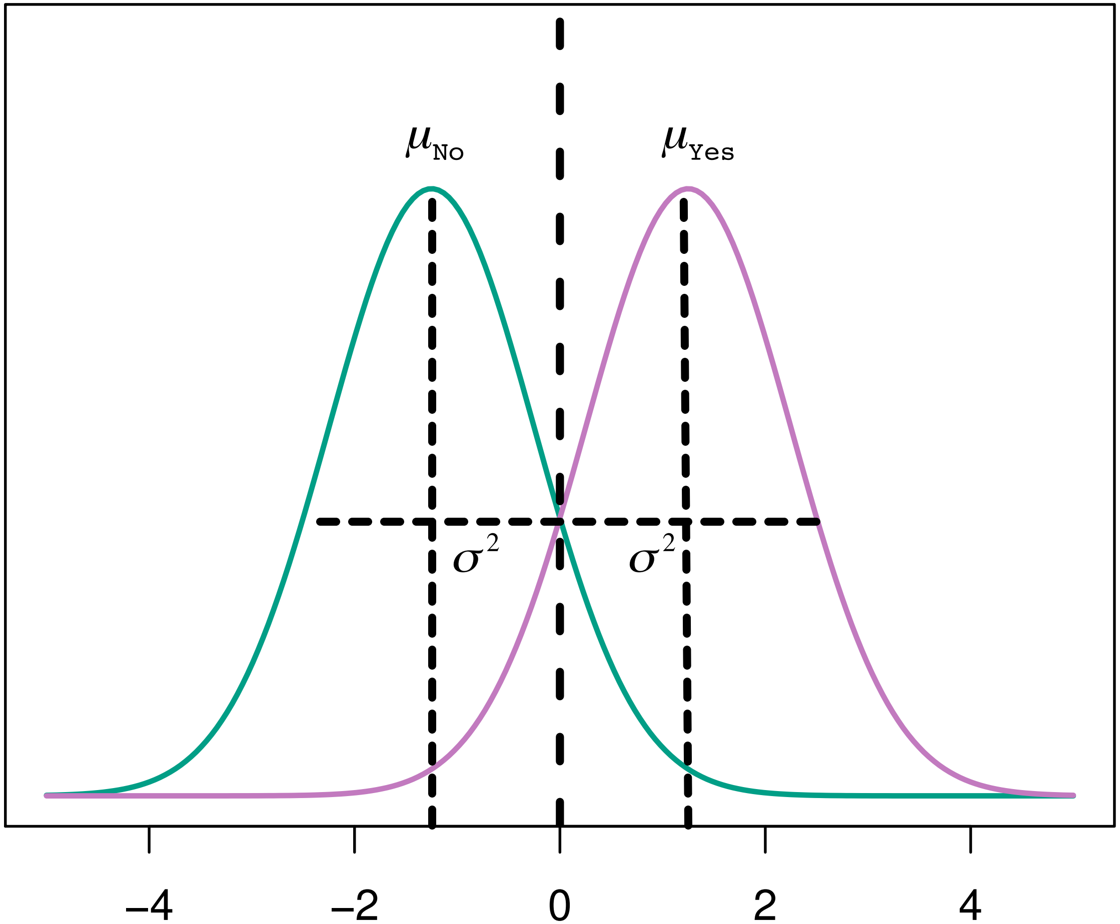
- For simplicity, we'll also assume that:

$$\sigma_1^2 = \cdots = \sigma_K^2$$

- That gives us a single variance term, which we'll denote

$$\sigma^2$$

Graphically



Plugging in…

$$p_k(x) = \frac{\Pr(Y=k) * \frac{1}{\sqrt{2\pi\sigma_k}} \times e^{-\frac{1}{2\sigma_k^2} * (x-\mu_k)^2}}{\sum_{i \in K} \Pr(Y=i) * \frac{1}{\sqrt{2\pi\sigma_i}} \times e^{-\frac{1}{2\sigma_i^2} * (x-\mu_i)^2}}$$

For our purposes, this is a constant!

More algebra!

- So really, we just need to maximize:

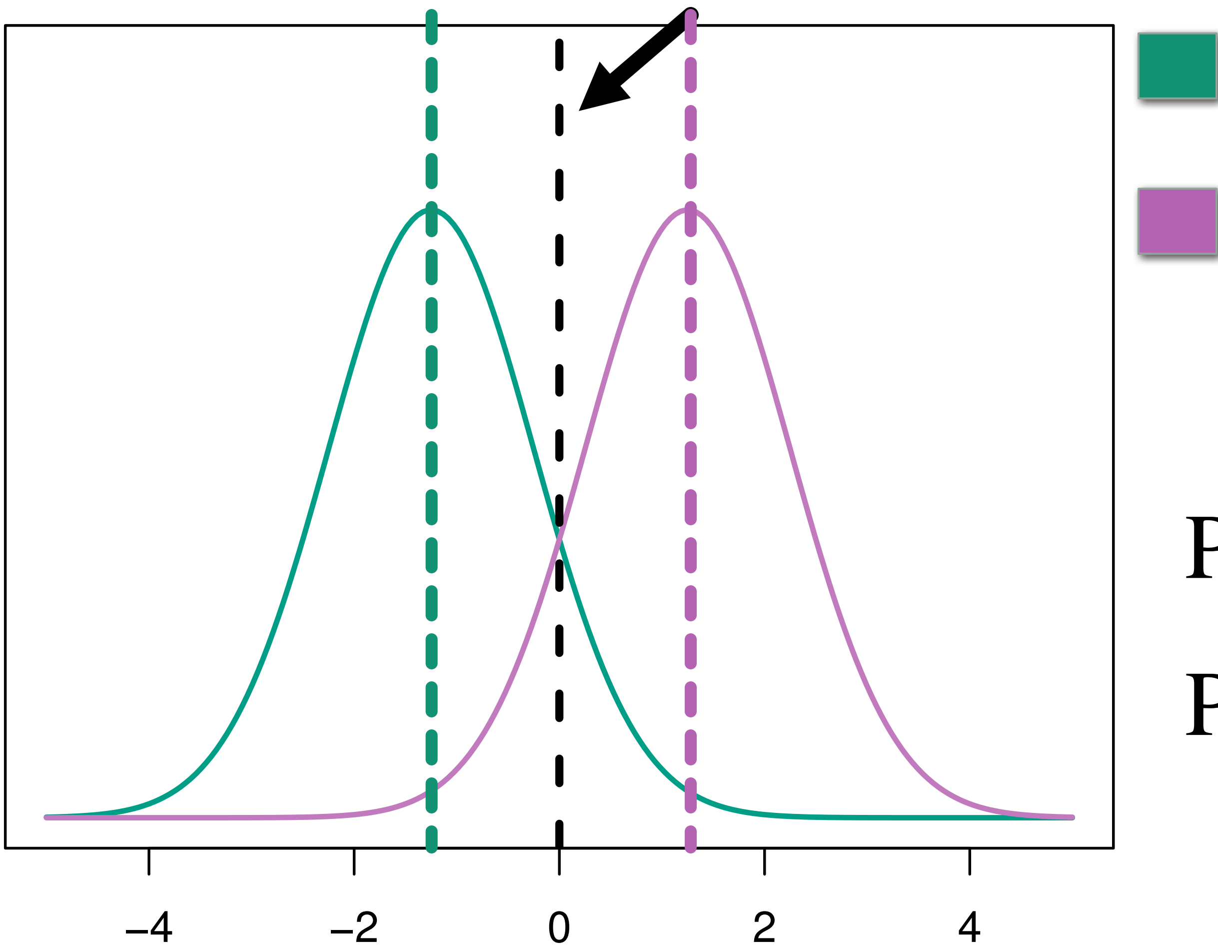
$$\Pr\left(Y=k\right)*\frac{1}{\sqrt{2\pi}\sigma_k}\times e^{-\frac{1}{2\sigma_k^2}*\left(x-\mu_k\right)^2}$$

$$\delta_k\left(x\right)=x*\frac{\mu_k}{\sigma^2}-\frac{\mu_k^2}{2\sigma^2}+\log\left(\Pr\left(Y=k\right)\right)$$

- This is called a *discriminant function* of x

Okay, we need an example

Bayes' Decision Boundary at *x=0*



$\mu_1 = -1.25$

$\mu_2 = 1.25$

$\sigma_1^2 = \sigma_2^2 = 1$

$\Pr(Y = 1) = 0.5$

$\Pr(Y = 2) = 0.5$

LDA: estimating the mean

- As usual, in practice we don't know the actual values for the parameters and so we have to estimate them

- The *linear discriminant analysis* method uses the following:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

(the average of all the training examples from class k)

LDA: estimating the variance

- Then we'll use that estimate to get:

$$\hat{\sigma} = \frac{1}{n-K} \sum_{K} \sum_{i:y_i=k} \left(x_i - \hat{\mu}_k \right)^2$$

(weighted average of the sample variances of each class)

Flashback

- Remember that time I said:

This isn't too hard to estimate*

$$\Pr(Y = k \mid X = x) = \frac{\Pr(X = x \mid Y = k) \times \Pr(Y = k)}{\Pr(X = x)}$$

- If we don't have additional knowledge about the class membership distribution:

$$\hat{\pi}_k = \frac{n_k}{n}$$

*Assuming our training data is randomly sampled

LDA classifier

- The LDA classifier plugs in all these estimates, and assign the observation to the class for which

$$\delta_k(x) = x * \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

is the largest

- The *linear* in LDA comes from the fact that this equation is linear in x

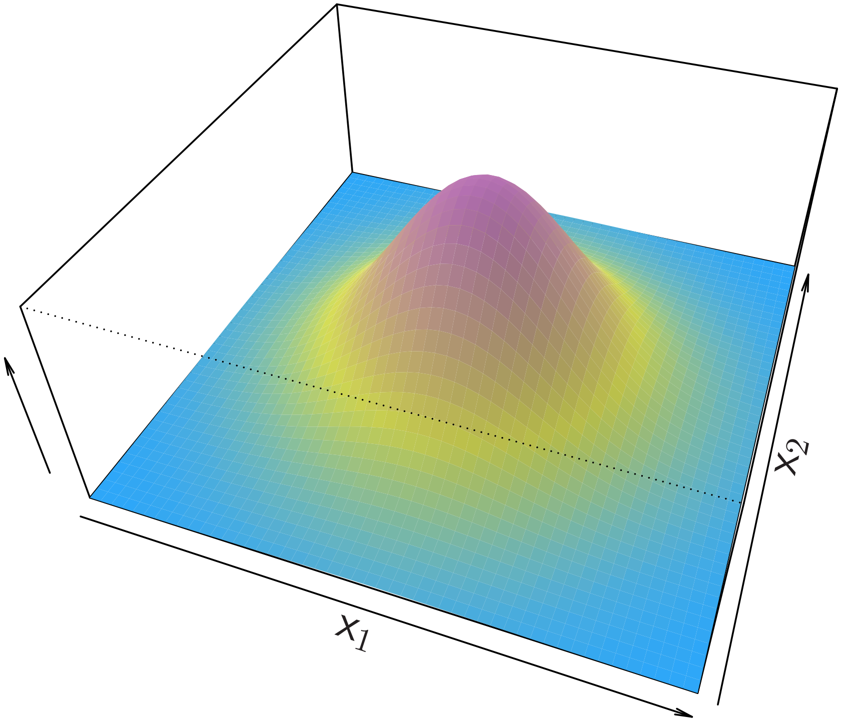
Quick recap: LDA

- LDA on 1 predictor makes 2 assumptions: what are they?
  1. Observations within class are normally distributed
  2. All classes have common variance

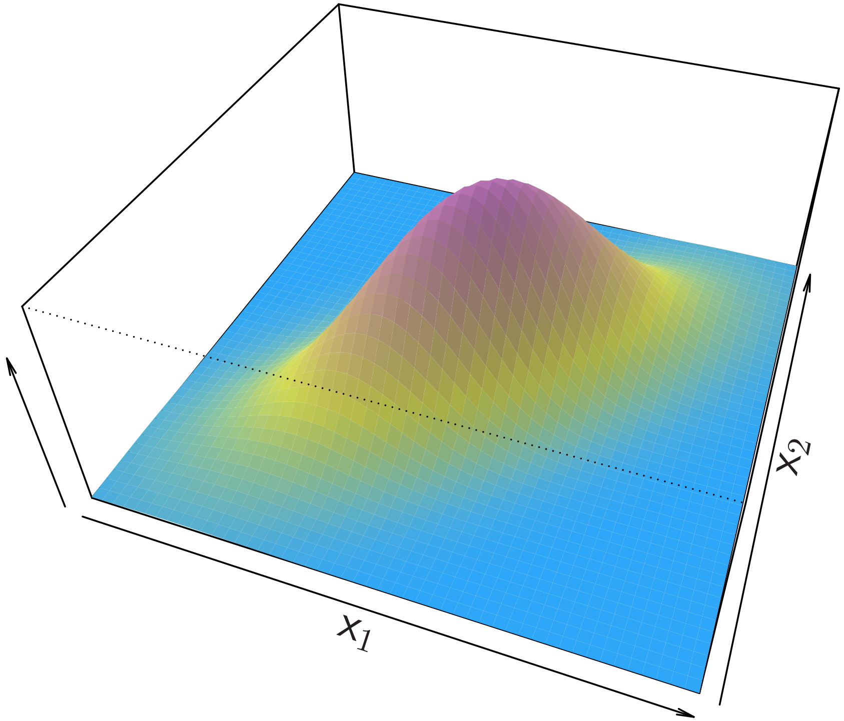
- So what would we need to change to make this work with **multiple** predictors?

LDA on multiple predictors

- Nothing! We just assume the density functions are *multivariate* normal:



uncorrelated



correlation = 0.7

LDA on multiple predictors

- Well okay, not **nothing**…

- What happens to the **mean**?

$$\mu_k : scalar \rightarrow vector$$

- What happens to the **variance**?

$$\sigma^2 : \; scalar \rightarrow Cov(\mathbf{X}) : matrix$$

Quick recap: LDA

- LDA on 1 predictor makes 2 assumptions: what are they?
  1. Each class is normally distributed
  2. All classes have common variance

What can we do about
This second assumption?

Quadratic discriminant analysis

- What if we relax the assumption that the classes have uniform variance?

$$Cov(\mathbf{X}) \rightarrow Cov_k(\mathbf{X}) \text{ for each } k$$

- If we plug **this** into Bayes', we get:

$$\delta_k(\vec{x}) = -\frac{1}{2}(\vec{x} - \vec{\mu}_k)^T Cov_k(\mathbf{X})^{-1}(\vec{x} - \vec{\mu}_k) + \log(\pi_k)$$

Multiplying two x terms together, hence "quadratic"

Discussion: QDA vs. LDA

- **Question:** why does it matter whether or not we assume that the classes have common variance?

- **Answer:** bias/variance trade off
  - One covariance matrix on p predictors → $p(p+1)/2$ parameters
  - The more parameters we have to estimate, the higher variance in the model (but the lower the bias)

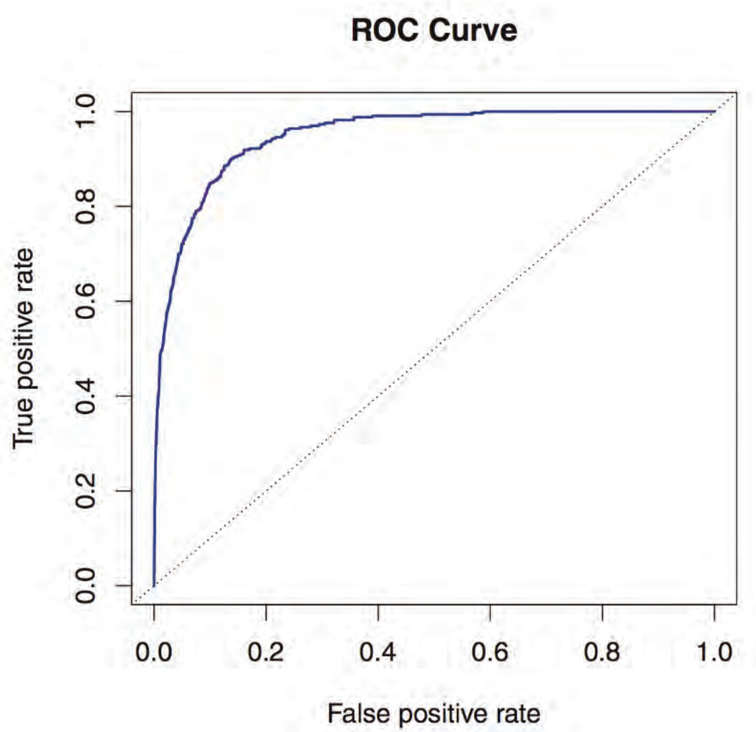
A diversion: Detection theory

- Detection theory is more traditionally considered in electrical engineering contexts
  - Radar, communications, biosurveillance sensors
- But the problem is the same as we consider here: discriminate signal from noise
  - Think of “signal” and “noise” as two classes
- The typical context, however, is different
  - Usually the classes are highly imbalanced

Class imbalance

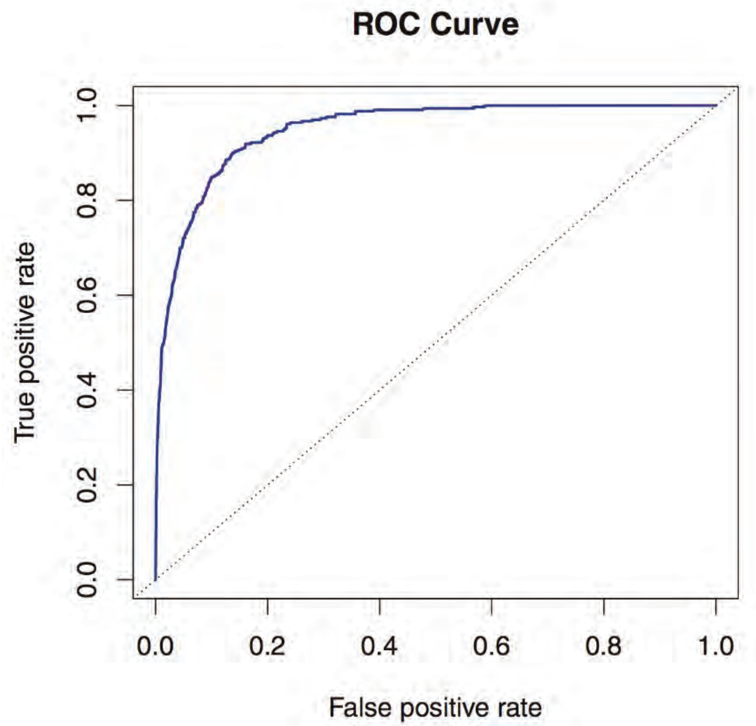
- Overall error rate is one metric for performance
- It doesn’t consider potential tradeoffs
- In a “detection” setting, only one class is interesting
  - Labeling something “noise” is saying, in effect, “I don’t care about this observation”
- In this scenario, when noise is misclassified as signal, it’s called a “false alarm” or “false positive”
  - Example: determining someone will default on their loan when they won’t

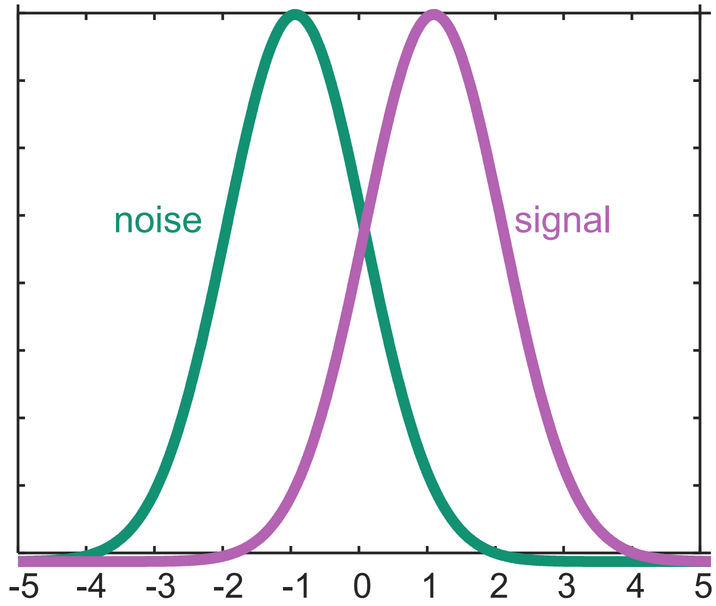
Receiver operating characteristic (ROC)



- ROC curves demonstrate the tradeoff between false positives and false negatives
- Perfect detection is achievable when the curves touches (1,1)
- Random assignment tracks the diagonal dotted line

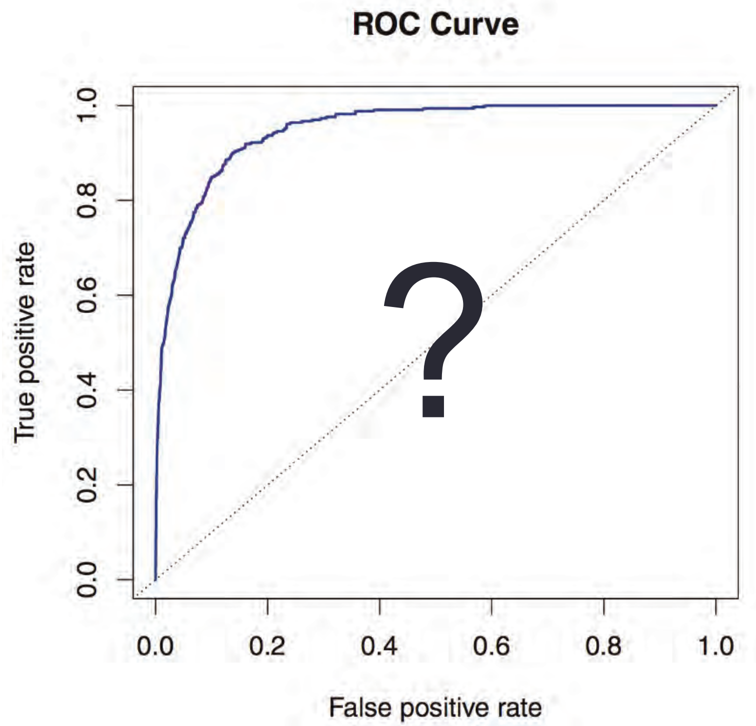
Class balance

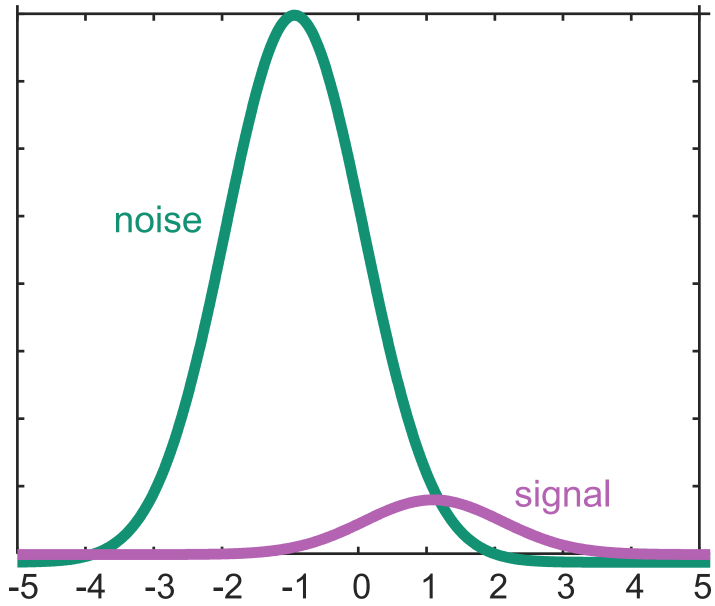




- Let's visually weight the pdfs by class size
- Suppose we have the ROC curve above for two balanced classes

Class balance

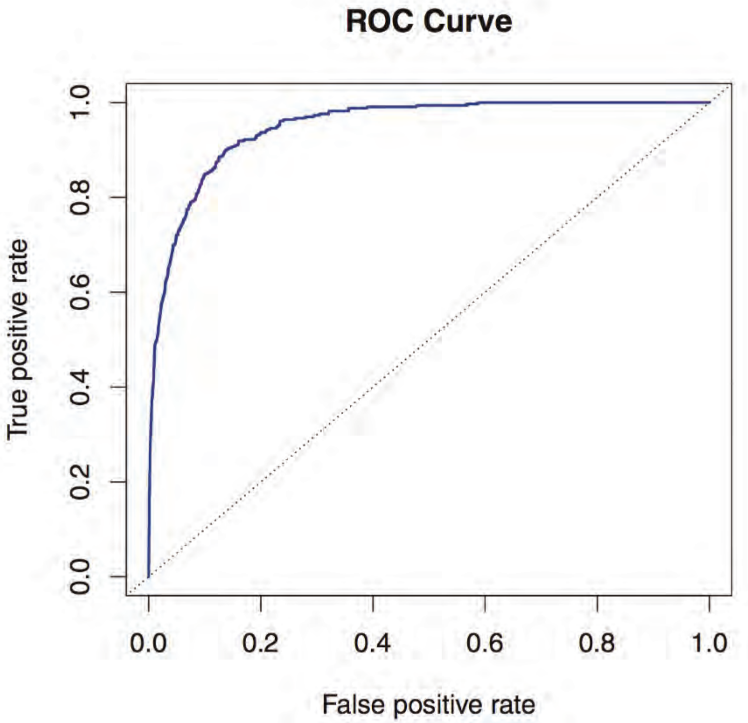




- Let's visually weight the pdfs by class size
- Suppose we have the ROC curve above for two balanced classes
- How will it change if the class balance changes?

ROC curves don't change with class balance!

$$TPR = \Pr(\text{classify as pos.} \mid \text{is pos.})$$
$$FPR = \Pr(\text{classify as pos.} \mid \text{is neg.})$$



- Probability of true positive and false positive are *conditioned* on the true state
- The conditional distributions don't change based on class balance
- (Aside: this is not the case for precision and recall, which you can read about in the book if you're interested)

Constant false alarm rate

- False alarms can be expensive
  - May require staff time to investigate a benign incident
- In a resource-constrained environment, a common practice is to set the *number* of false alarms that can be tolerated
- So what should we do?

Constrained optimization for detection

- Datapoint x: how many measurements to you get per unit time?
  - E.g., credit applications per day, radar returns per second
- Datapoint y: how many false alarms can you tolerate per unit time?
- Worst case scenario: there's nothing to detect; any detections are false alarms
- Set false positive rate to y/x
- Now you have your false alarm rate, so *maximize your detection rate*

Maximizing the probability of detection

- There is a certain *region* of the measurement space where we will declare a detection
  - Everywhere else we won't
  - $TPR = \int_{x \in R} p_1(x)\,dx$
  - $FPR = \int_{x \in R} p_2(x)\,dx$
  - Want to maximize $TPR + \lambda(FPR - \alpha)$, where α is the desired false alarm rate
    - (Trust me for now, but if you're curious as to why, come ask after the lecture)

Maximizing the probability of detection

- It turns out we don't even have to do any calculus!
- $$\begin{aligned} & TPR + \lambda(FPR - \alpha) \\ &= \left(\int_{x \in R} p_1(x)\,dx\right) + \lambda\left(\int_{x \in R} p_2(x)\,dx - \alpha\right) \\ &= \int_{x \in R} (p_1(x) + \lambda p_2(x))\,dx - \lambda\alpha \end{aligned}$$
- No restrictions exist on the region
- To maximize this quantity set R to everywhere the integrand is positive!
- $p_1(x) + \lambda p_2(x) > 0 \Longrightarrow p_1(x)/p_2(x) > -\lambda$
- (Then set λ to achieve the desired FPR)

Q: Why am I talking about this here?

- A1: just to provide a different perspective
- A2: Because the same mathematical principles are at work as with LDA and QDA
- What's the optimal detector for two Gaussians?
- $\frac{p_1(x)}{p_2(x)} > c \Longrightarrow \frac{\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{(2\sigma_1^2)}}}{\frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x-\mu_2)^2}{(2\sigma_2^2)}}} > c \Longrightarrow \frac{\sigma_2}{\sigma_1} e^{\frac{(x-\mu_2)^2}{(2\sigma_2^2)} - \frac{(x-\mu_1)^2}{(2\sigma_1^2)}} > c$

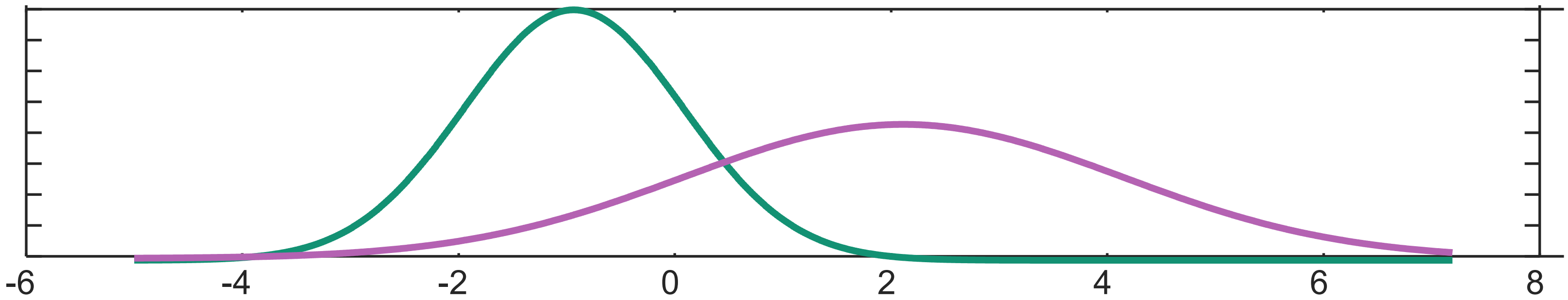
$$\Longrightarrow \ln \sigma_2 - \ln \sigma_1 + \frac{(x-\mu_2)^2}{(2\sigma_2^2)} - \frac{(x-\mu_1)^2}{(2\sigma_1^2)} > \ln c$$

Q: Why am I talking about this here?

- What happens if the variances are the same?
- $$\ln\sigma_2 - \ln\sigma_1 + \frac{(x-\mu_2)^2}{(2\sigma_2^2)} - \frac{(x-\mu_1)^2}{(2\sigma_1^2)} > \ln c$$
  $$\Longrightarrow 2(\mu_1 - \mu_2)x + (\mu_2^2 - \mu_1^2 - 2\sigma^2 \ln c) > 0$$
  - A line optimally discriminates between signal and noise!
- And if they're different?
- $$\ln\sigma_2 - \ln\sigma_1 + \frac{(x-\mu_2)^2}{(2\sigma_2^2)} - \frac{(x-\mu_1)^2}{(2\sigma_1^2)} > \ln c \Longrightarrow$$
  $$(\sigma_1^2 - \sigma_2^2)x^2 + 2(\sigma_2^2\mu_1 - \sigma_1^2\mu_1)x$$
  $$+\left(\sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln\frac{\sigma_2}{c\sigma_1}\right) > 0$$
  - Use a quadratic!
- This generalizes to higher dimensions

This can be a little counterintuitive

- If the variance is different for signal and noise, a threshold is not the best you can do
- Consider these distributions:



- The variance of the signal is greater than the variance of the noise
  - So there are actually values that are so *small* they should be classified as signal!
    - If the signal variance were smaller, there would be values so *large* they should be classified as noise
  - (These are often so far in the tails that they're extremely unlikely)

Lab: LDA and QDA

- To do today's lab in R: nothing new
- To do today's lab in python: nothing new
- Instructions and code:

  [course website]/labs/lab5-r.html

  [course website]/labs/lab5-py.html

- You'll notice that the labs are beginning to get a bit more open-ended – this may cause some discomfort!
  - **Goal:** to help you become more fluent in using these methods
  - Ask questions early and often: of me, of each other, of whatever resource helps you learn!
- Full version can be found beginning on p. 161 of ISLR

Coming up

- Wednesday– last day of classification: comparing methods
- Solutions for A1 have been posted to the course website and Moodle
- A2 due on **Wednesday Oct. 4th by 11:59pm**