Introduction to Machine Learning – Generative Models

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (https://jcrouser.github.io/)

Plan for Today

- Bayes Classifier
- Linear Discriminant Analysis
- Classification Errors

Warm Up: Logistic Regression

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fctr>
1	1990	0.816	1.572	-3.936	-0.229	-3.484	0.1549760	-0.270	Down
2	1990	-0.270	0.816	1.572	-3.936	-0.229	0.1485740	-2.576	Down
3	1990	-2.576	-0.270	0.816	1.572	-3.936	0.1598375	3.514	Up
4	1990	3.514	-2.576	-0.270	0.816	1.572	0.1616300	0.712	Up
5	1990	0.712	3.514	-2.576	-0.270	0.816	0.1537280	1.178	Up
6	1990	1.178	0.712	3.514	-2.576	-0.270	0.1544440	-1.372	Down

```
Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Weekly)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6949  -1.2565   0.9913   1.0849   1.4579

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.26686    0.08593   3.106   0.0019 **
Lag1        -0.04127    0.02641  -1.563   0.1181
Lag2         0.05844    0.02686   2.175   0.0296 *
Lag3        -0.01606    0.02666  -0.602   0.5469
Lag4        -0.02779    0.02646  -1.050   0.2937
Lag5        -0.01447    0.02638  -0.549   0.5833
Volume      -0.02274    0.03690  -0.616   0.5377
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Which predictors are significant?

Warm Up: Logistic Regression

	Year <dbl>	Lag1 <dbl>	Lag2 <dbl>	Lag3 <dbl>	Lag4 <dbl>	Lag5 <dbl>	Volume <dbl>	Today <dbl>	Direction <fctr>
1	1990	0.816	1.572	-3.936	-0.229	-3.484	0.1549760	-0.270	Down
2	1990	-0.270	0.816	1.572	-3.936	-0.229	0.1485740	-2.576	Down
3	1990	-2.576	-0.270	0.816	1.572	-3.936	0.1598375	3.514	Up
4	1990	3.514	-2.576	-0.270	0.816	1.572	0.1616300	0.712	Up
5	1990	0.712	3.514	-2.576	-0.270	0.816	0.1537280	1.178	Up
6	1990	1.178	0.712	3.514	-2.576	-0.270	0.1544440	-1.372	Down

Confusion Matrix for our model:

```
glm.pred Down  Up
    Down   54  48
    Up    430 557
```

Actual

Predicted

What is the overall error rate for our model's predictions?

*More about error later!

Generative Models

- Logistic Regression directly models $\Pr(Y = k|X = x)$
  - i.e., we model the conditional distribution of Y given the predictor(s) X
- Alternatively, we can model the distribution of predictors, X, separately for each response class. Then use Bayes Theorem to flip them into estimates for $\Pr(Y = k|X = x)$

Generative Models

Why bother?

- When there is substantial separation between the two classes, Logistic Regression parameter estimates are unstable
- If the distribution of the predictors is approximately normal in each class and the sample size is small Generative Models will tend to outperform Logistic Models
- Generative Models extend to more than two classes much more seamlessly

Generative Models

Bayes Theorem

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- π_k is the ***prior probability*** that a randomly chosen observation come from the kth class
- $f_k(X) \equiv \Pr(X|Y = k)$ is the ***density function*** of X for an observation that comes from the kth class
- $\Pr(Y = k|X = x)$ is the ***posterior probability***, i.e. the probability that an observation belongs to the kth class given the predictor value *(x)* for the observation

Generative Models

Easy to estimate! How?

Bayes Theorem

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- π_k is the ***prior probability*** that a randomly chosen observation come from the kth class
- $f_k(X) \equiv \Pr(X|Y = k)$ is the ***density function*** of X for an observation that comes from the kth class
- $\Pr(Y = k|X = x)$ is the ***posterior probability***, i.e. the probability that an observation belongs to the kth class given the predictor value *(x)* for the observation

Generative Models

Easy to estimate! How?

Hard to estimate, many options

Bayes Theorem

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- π_k is the ***prior probability*** that a randomly chosen observation come from the kth class
- $f_k(X) \equiv \Pr(X|Y = k)$ is the ***density function*** of X for an observation that comes from the kth class
- $\Pr(Y = k|X = x)$ is the ***posterior probability***, i.e. the probability that an observation belongs to the kth class given the predictor value *(x)* for the observation

Generative Models

Easy to estimate!
How?

Bayes Theorem

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- π_k is the ***prior probability*** that a randomly chosen observation come from the kth class
- $f_k(X) \equiv \Pr(X|Y = k)$ is the ***density function*** of X for an observation that comes from the kth class
- $\Pr(Y = k|X = x)$ is the ***posterior probability***, i.e. the probability that an observation belongs to the kth class given the predictor value *(x)* for the observation

Hard to estimate, many options
If we had the **perfect** f, then we'd have a Bayes classifier

LDA

For generative models, we need to estimate $f_k(x)$ to plug into $\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$

Linear Discriminant Analysis (LDA) with one predictor (p=1)

Assumption 1:

- $f_k(x)$ is *normal* or *Gaussian*
- Then, $f_k(x) = \frac{1}{\sqrt{2\pi_k \sigma_k}} * e^{\left(\frac{-1}{2\sigma_k^2}(x-\mu_k)^2\right)}$

where μ_k and σ_k^2 are the mean and variance of the kth class

LDA

For generative models, we need to estimate $f_k(x)$ to plug into $\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$

Linear Discriminant Analysis (LDA) with one predictor (p=1)

Assumption 2:

- classes have equal variance, i.e.
  - $\sigma_1^2 = \ ... = \sigma_k^2$

So we can use one variance term, σ^2

LDA

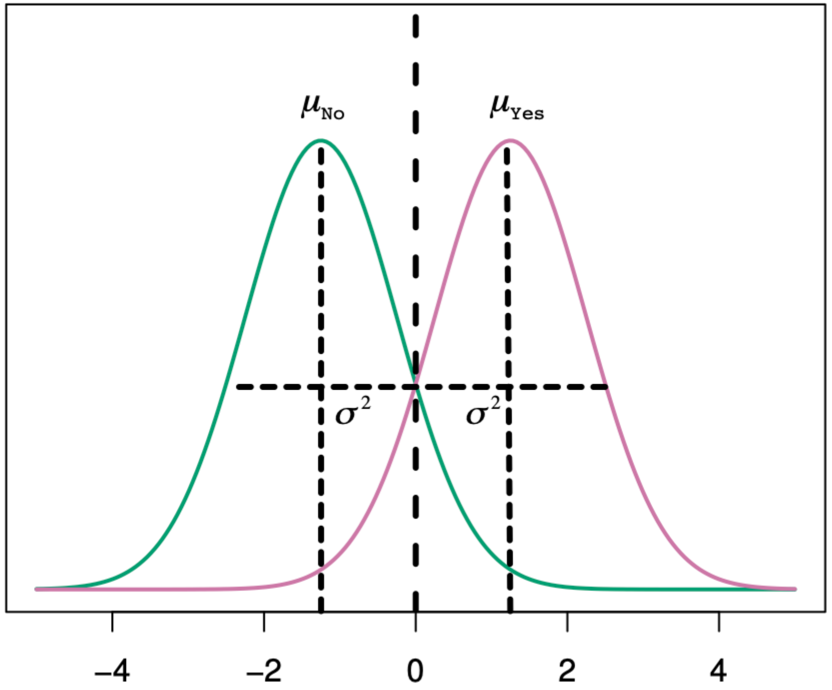
For generative models, we need to estimate $f_k(x)$ to plug into $\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$

Linear Discriminant Analysis (LDA) with one predictor (p=1)

Assumption 2:

- classes have equal variance, i.e.
  - $\sigma_1^2 = \ldots = \sigma_k^2$

So we can use one variance term, σ^2



LDA

For generative models, we need to estimate $f_k(x)$ to plug into $\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$

Linear Discriminant Analysis (LDA) with one predictor (p=1)

Assumption 2:

- classes have equal variance, i.e.
  - $\sigma_1^2 = \ldots = \sigma_k^2$

  So we can use one variance term, σ^2

  Then,

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

LDA

For generative models, we need to estimate $f_k(x)$ to plug into $\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$

Linear Discriminant Analysis (LDA) with one predictor (p=1)

Assumption 2:

- classes have equal variance, i.e.
  - $\sigma_1^2 = \ \ldots = \sigma_k^2$

  So we can use one variance term,
  σ^2
  Then,

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

For our purposes, this is a constant

LDA

For generative models, we need to estimate $f_k(x)$ to plug into $\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$

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

For our purposes, this is a constant

So we really just need to maximize:

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

LDA

So we really just need to maximize:

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

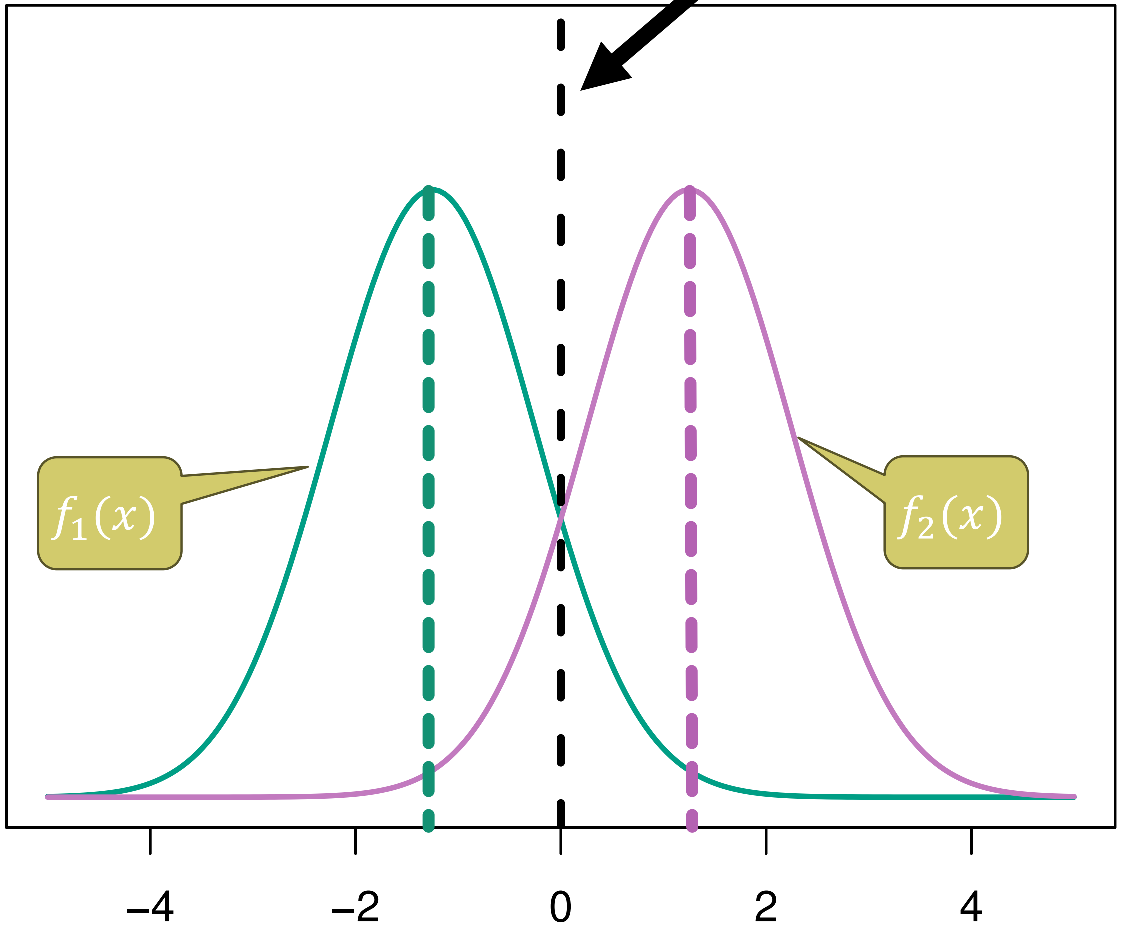
$$\delta_k(x) = x * \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\Pr(Y = k))$$

This is called a ***discriminant function*** of x

Example

LDA

Bayes' Decision Boundary at $x=0$



$\mu_1 = -1.25$

$\mu_2 = 1.25$

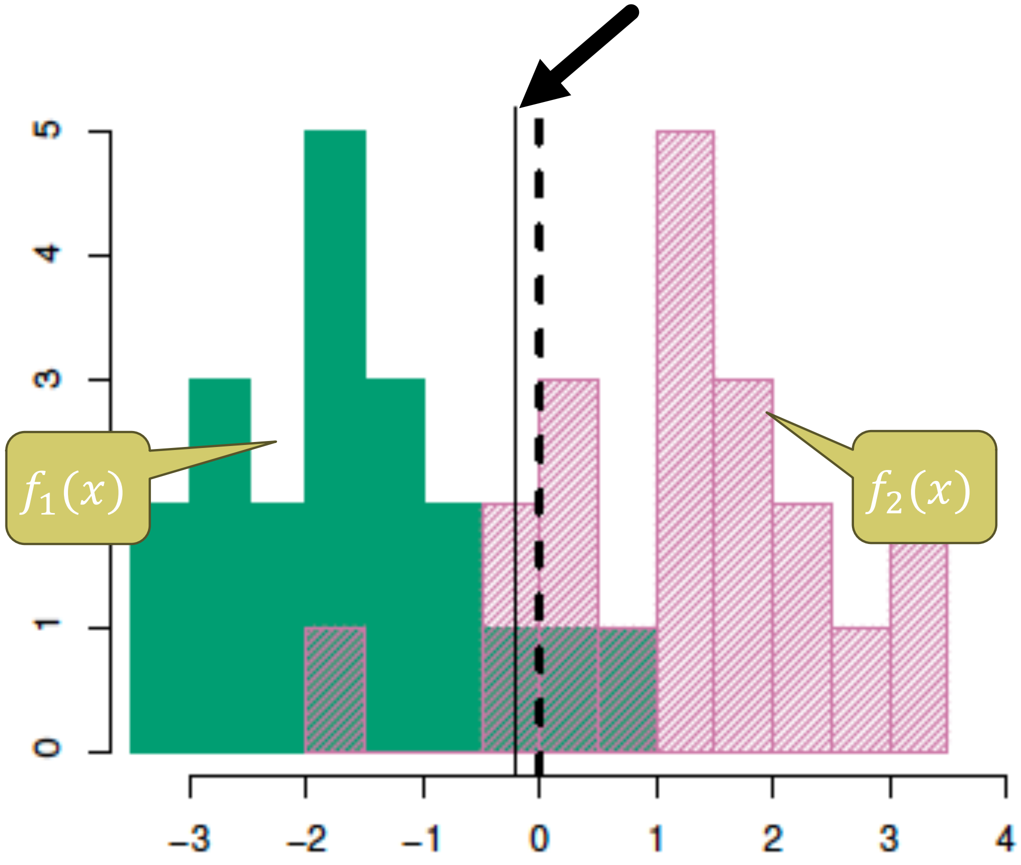
$\sigma_1^2 = \sigma_2^2 = 1$

$\Pr(Y = 1) = 0.5$

$\Pr(Y = 2) = 0.5$

LDA

LDA Decision Boundary



$$\mu_1 = -1.25$$

$$\mu_2 = 1.25$$

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

$$\Pr(Y = 1) = 0.5$$

$$\Pr(Y = 2) = 0.5$$

The decision boundary is where $\delta_1(x) = \delta_2(x)$

x values to the left of the boundary are assigned to green, and to the right are assigned to purple

Estimating parameters

LDA

- In practice, we don't know the actual values for the parameters, so we have to estimate them

- The LDA method uses the following estimate:

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

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

Estimating parameters

LDA

- In practice, we don't know the actual values for the parameters, so we have to estimate them

- The LDA method uses the following estimate:

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

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

- Using that mean estimate, we then get:

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

(weighted average of the sample variances of each class)

LDA

Estimating parameters

- In practice, we don't know the actual values for the parameters, so we have to estimate them

- The LDA method uses the following estimate:

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

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

- Using that mean estimate, we then get:

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

(weighted average of the sample variances of each class)

- And remember, we estimate $\hat{\pi}_k = \frac{n_k}{n}$

LDA

LDA uses all of those estimates to get the discriminant function, and assigns an observation to the class for which the function is largest.

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

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

The linear in LDA comes form the fact that this function is linear in x

LDA

LDA on one predictor makes 2 assumptions:

- Observations within class are normally distributed
- All classes have common variance

What do you think we need to change to work with **multiple** predictors?

LDA

LDA on one predictor makes 2 assumptions:

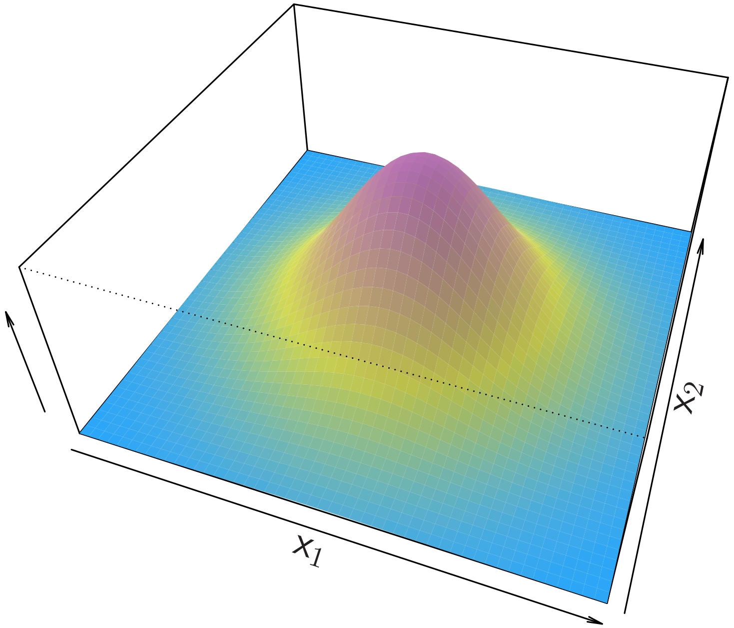
- Observations within class are normally distributed
- All classes have common variance

What do you think we need to change to work with **multiple** predictors?

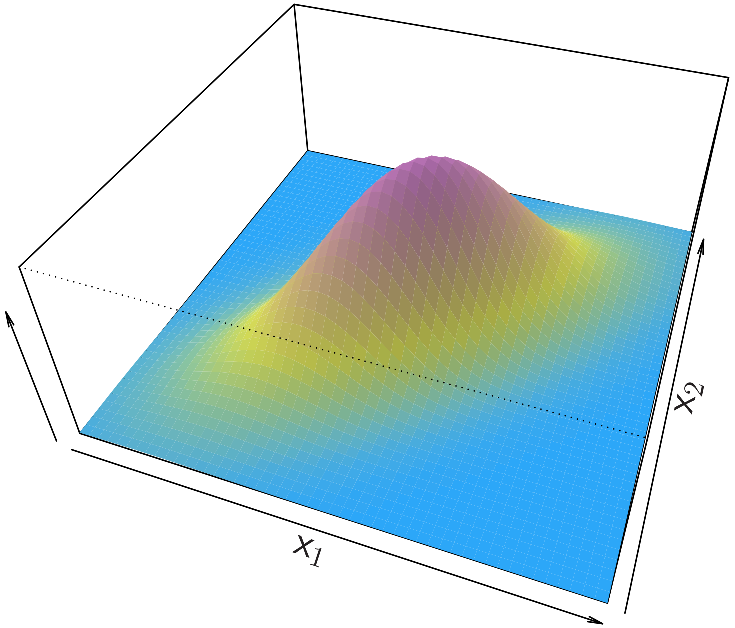
LDA

LDA on *multiple predictors*

- Assume observations within class are *multivariate normally distributed*



uncorrelated



correlation or unequal variance

LDA

LDA on *multiple predictors*

- Assume observations within class are *multivariate normally distributed*

  - What happens to the **mean**?

    $\mu_k: scalar \ \rightarrow vector$ (with p components)

  - What happens to the **variance**?

    $\sigma^2: scalar \ \rightarrow \Sigma: matrix$ (p x p covariance matrix of X)

LDA

LDA on one predictor makes 2 assumptions:

- Observations within class are normally distributed
- All classes have common variance

LDA

LDA on *multiple predictors*

- Assume observations within class are *multivariate normally distributed*

  - What happens to the **mean**?

$$\mu_k : scalar \;\rightarrow vector \text{ (with p components)}$$

  - What happens to the **variance**?

$$\sigma^2 : scalar \;\rightarrow \Sigma : matrix \text{ (p x p covariance matrix of X)}$$

Assume equal for all classes, K

LDA

LDA on *multiple predictors*

- Assume observations within class are *multivariate normally distributed*

  - What happens to the **mean**?

    $$\mu_k: scalar \ \rightarrow vector \text{ (with p components)}$$

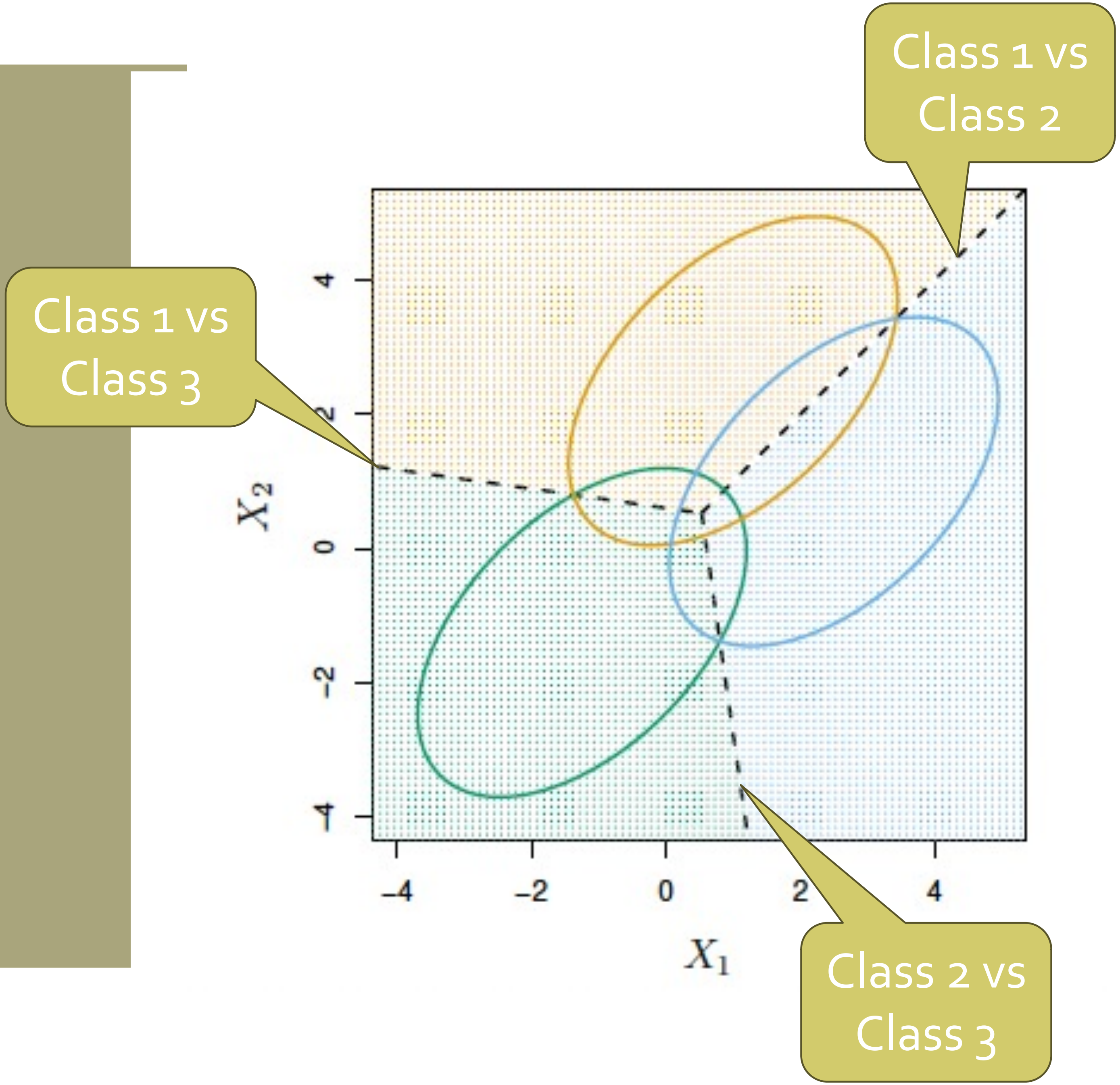
  - What happens to the **variance**?

    $$\sigma^2: scalar \ \rightarrow \Sigma: matrix \text{ (p x p covariance matrix of X)}$$

- Plugging in, we get the matrix version of our previous equation:

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

LDA

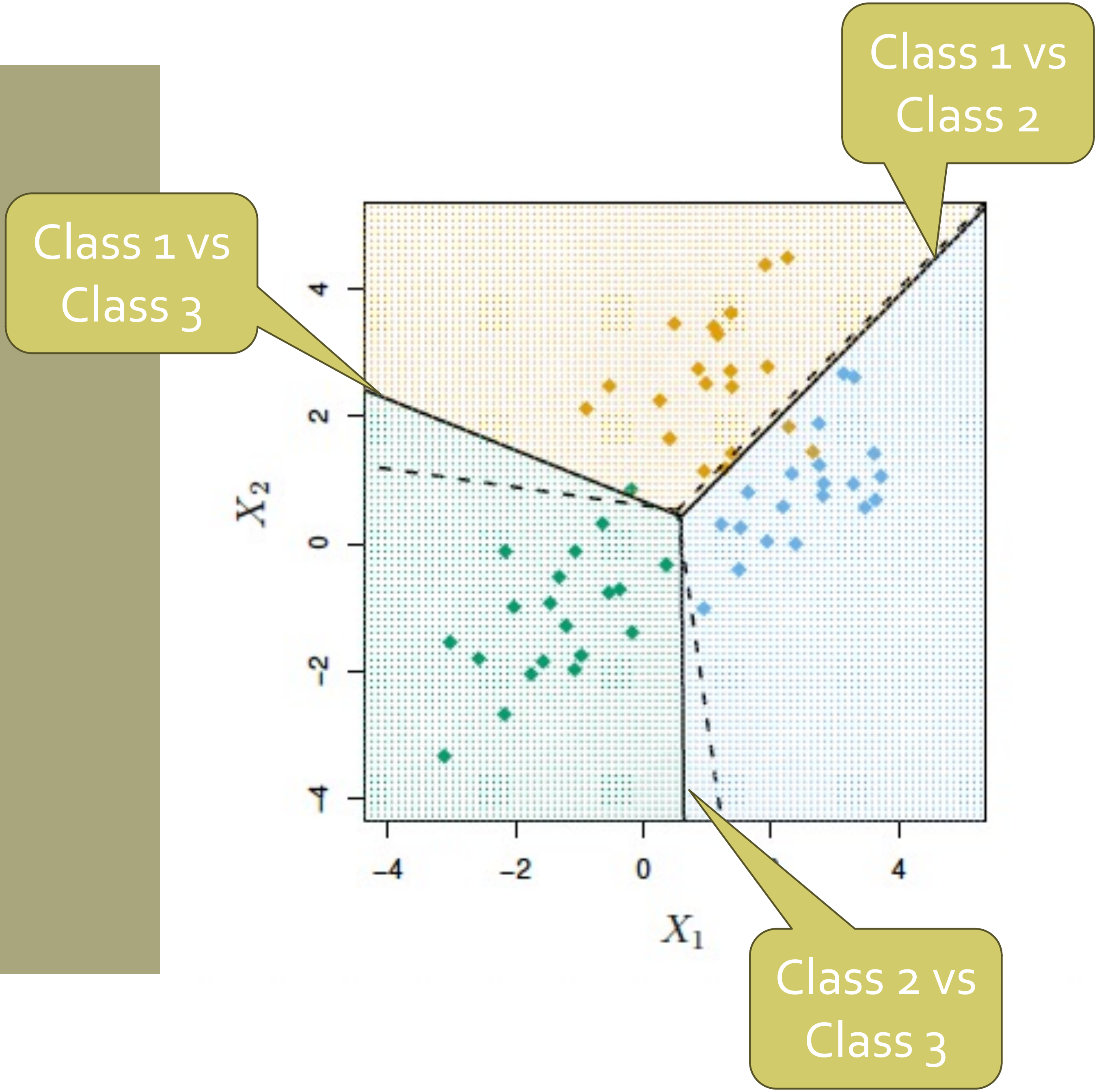


p = 2

Ellipses represent 95% of the probability for each class

Dashed lines are the Bayes decision boundaries

LDA



p = 2

Ellipses represent 95% of the probability for each class

Dashed lines are the Bayes decision boundaries

Solid lines are LDA decision boundaries

Classification Error

LDA on `Default` data (from logistic regression lecture)

Confusion matrix on *training data*

		True default status		
		No	Yes	Total
Predicted default status	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

What is the overall error rate? Does that seem good or bod?
Hint: Think about the error rate you'd get just from saying no one defaults.

Classification Error

LDA on `Default` data (from logistic regression lecture)

Confusion matrix on *training data*

		True default status		
		No	Yes	Total
Predicted default status	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

There are two types (or categories) of error here. What are they?

Classification Error

LDA on `Default` data (from logistic regression lecture)

Confusion matrix on *training data*

		True default status		
		No	Yes	Total
Predicted default status	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

False Negative

False Positive

Does the model perform equally well within each error type?

Classification Error

LDA on `Default` data (from logistic regression lecture)

Confusion matrix on *training data*

		True default status		
		No	Yes	Total
Predicted default status	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

False Negative

False Positive

- ***sensitivity*** refers to the percent of true positives
- ***specificity*** refers to the percent of true negatives

What is the sensitivity and specificity of our model?

Classification Error

LDA on `Default` data (from logistic regression lecture)

Confusion matrix on *training data*

		True default status		
		No	Yes	Total
Predicted	No	9644	252	9896
default status	Yes	23	81	104
	Total	9667	333	10000

False Negative

False Positive

- ***sensitivity*** refers to the percent of true positives
- ***specificity*** refers to the percent of true negatives

How could we make our model more sensitive? Ideas?

Classification Error

Increasing sensitivity of LDA

- Remember this?
  - $\Pr(Y = k|X = x)$ is the ***posterior probability***, i.e. the probability that an observation belongs to the kth class given the predictor value *(x)* for the observation
- Bayes Classifiers use a posterior probability of 0.5 (for two classes)
  - In the `Default` example we assign an observation to default if
$$\Pr(default = Yes \mid X = x) > 0.5$$
- However, we can lower this threshold!
  - Ex. we can assign an observation to default if
$$\Pr(default = Yes \mid X = x) > 0.2$$

Classification Error

LDA on `Default` data with new posterior probability

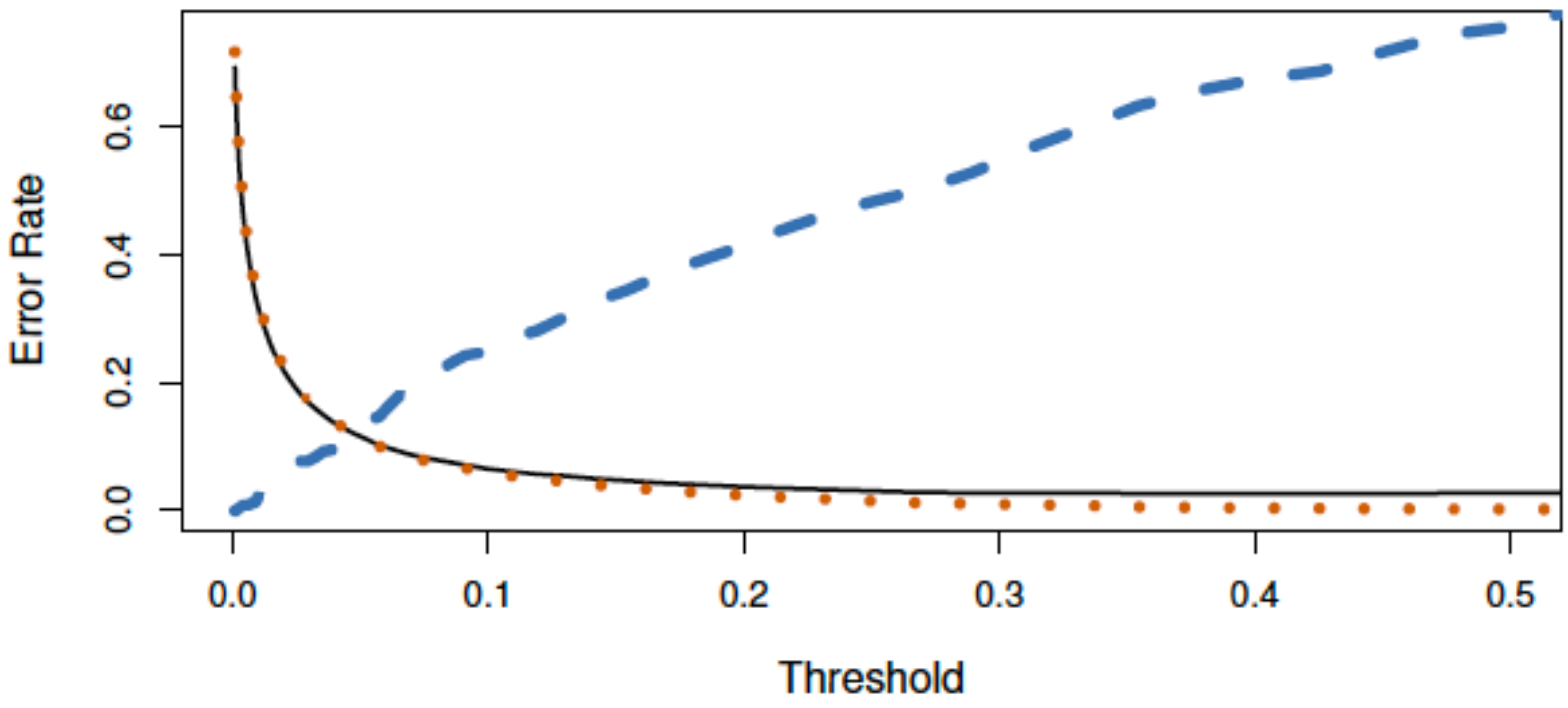
Confusion matrix on *training data*

		True default status		
		No	Yes	Total
Predicted default status	No	9432	138	9570
	Yes	235	195	430
	Total	9667	333	10000

Is this better?

Classification Error

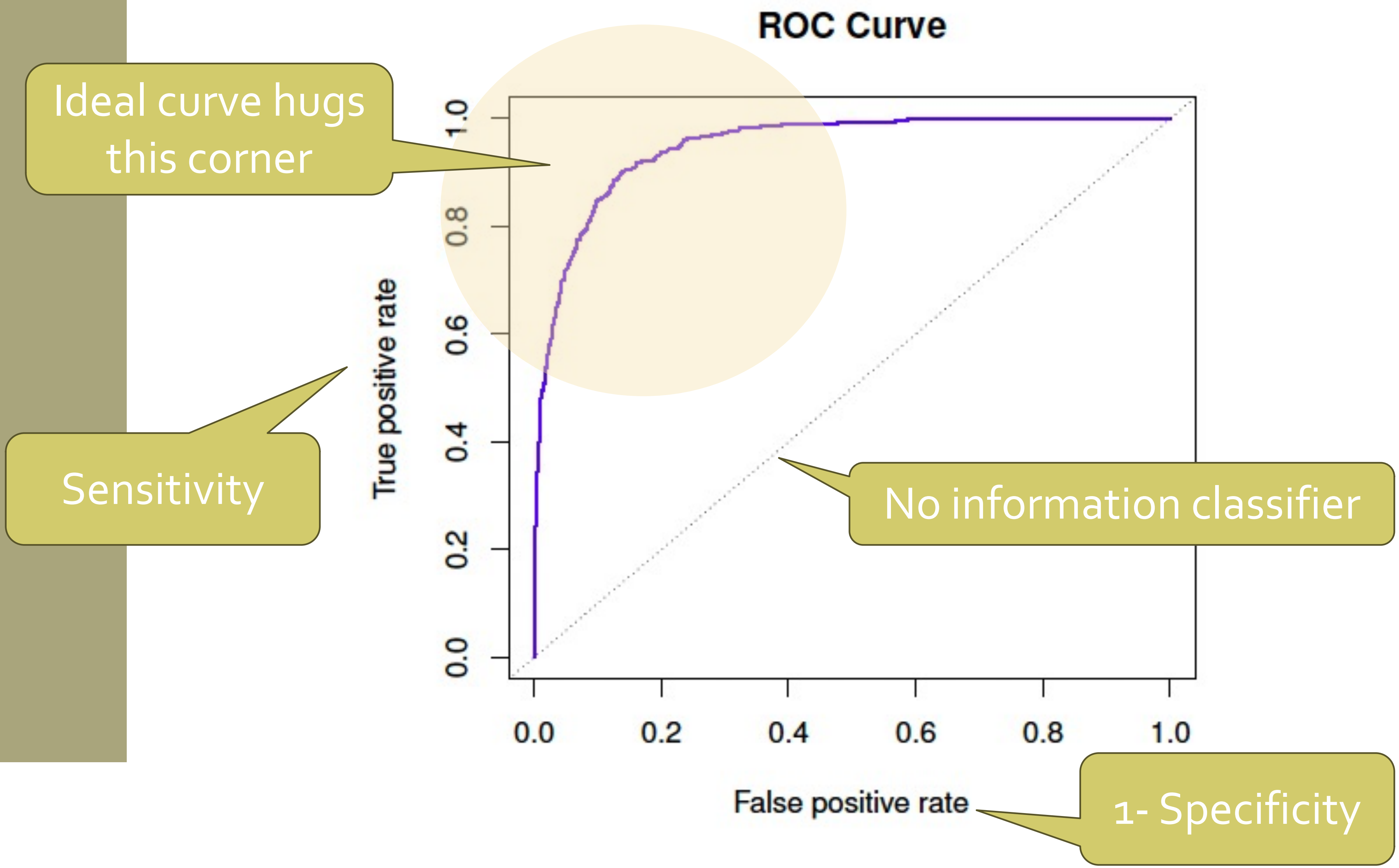
Tradeoff when modifying posterior probability



- Black solid line = overall error
- Blue dashed line = False negatives
- Orange dotted line = False positives

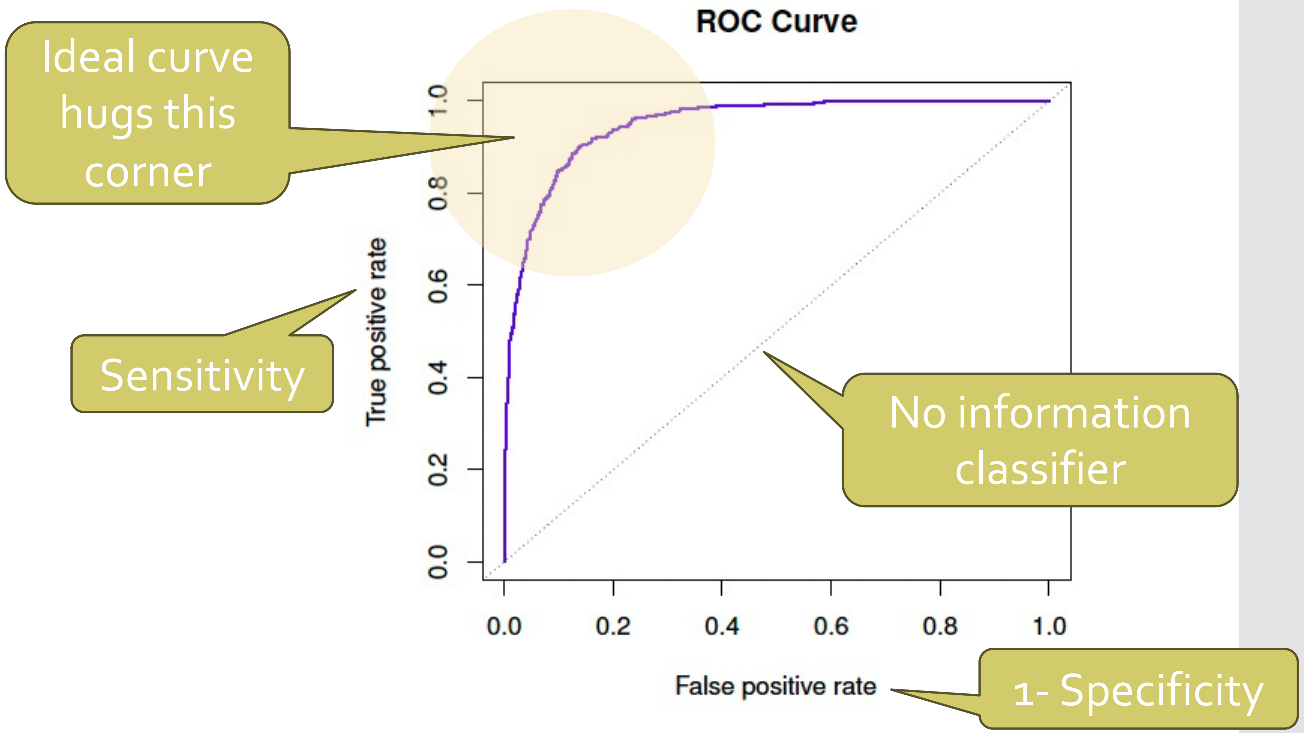
ROC Curve

ROC Curve for LDA



ROC Curve

AUC (Area Under the ROC Curve)



- The overall performance of a classifier summarized over all possible thresholds is the AUC
- Maximum is 1, so numbers closer to that are better
- Useful way to compare difference classifiers

Error Terms

		True class		
		− or Null	+ or Non-null	Total
Predicted class	− or Null	True Neg. (TN)	False Neg. (FN)	N*
	+ or Non-null	False Pos. (FP)	True Pos. (TP)	P*
	Total	N	P	

Error Terms

		True class		
		− or Null	+ or Non-null	Total
Predicted class	− or Null	True Neg. (TN)	False Neg. (FN)	N*
	+ or Non-null	False Pos. (FP)	True Pos. (TP)	P*
	Total	N	P	

Consider our confusion matrix from our logistic model. What is a false positive? What is a false negative? What are the false positive and false negative rates? What are the sensitivity and specificity of the model?

Actual

```
glm.pred Down  Up
    Down   54  48
    Up    430 557
```

Predicted