Lecture 4: Classification

STAT 1361/2360: Statistical Learning and Data Science

University of Pittsburgh
Prof. Lucas Mentch

Linear Regression

- In Chapter 3, we looked at Linear Regression:
  - Data of the form (x_i, y_i) for $i = 1, \ldots, n$ where $x_i = (x_{1,i}, \ldots, x_{p,i})$ and $y_i \in \mathbb{R}$
  - Assume

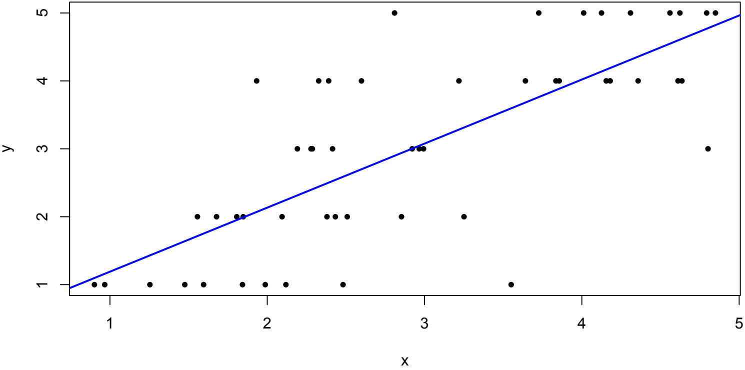
$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

  - What if $y_i \in \{a_1, a_2, \ldots, a_m\}$? (i.e. Y is categorical) Can we still use linear regression?

Yes, sometimes, but only in certain situations and always with substantial caution

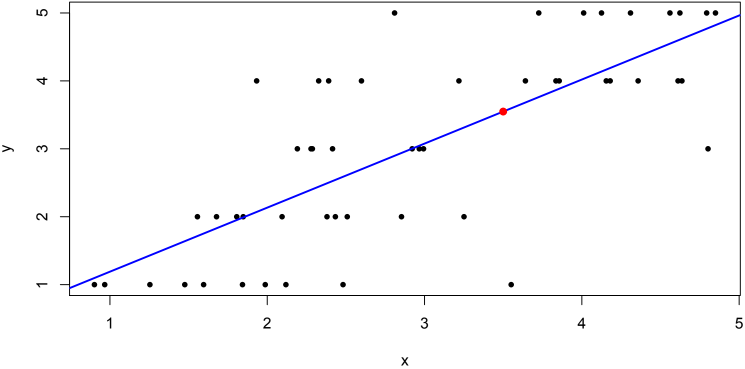
Linear Regression with Categorical Response

Suppose we have an ordinal response $Y \in \{1, 2, 3, 4, 5\}$ where, for example, 1 = “Poor", ..., 5 = “Great"



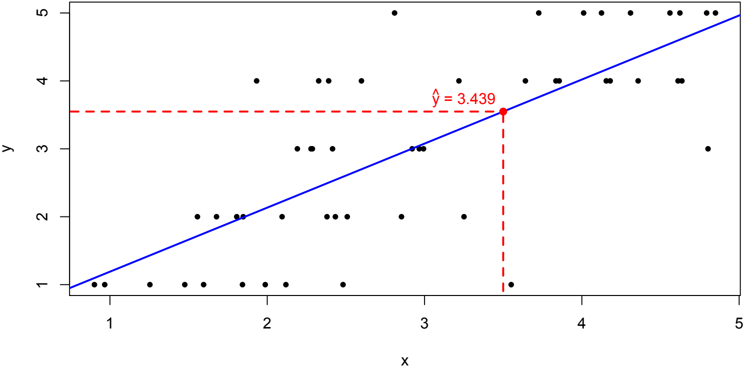
Linear Regression with Categorical Response

Suppose we have an ordinal response $Y \in \{1, 2, 3, 4, 5\}$ where, for example, 1 = “Poor", ..., 5 = “Great"



Linear Regression with Categorical Response

Suppose we have an ordinal response $Y \in \{1, 2, 3, 4, 5\}$ where, for example, 1 = “Poor", ..., 5 = “Great"



Linear Regression with Categorical Response

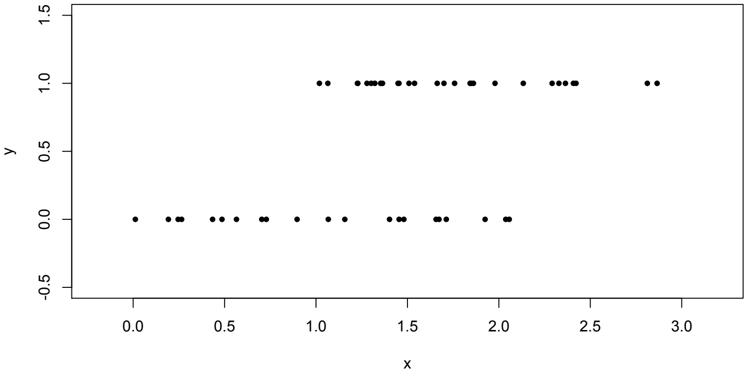
- If we fit a linear model to data like this, our predictions are almost always going to fall “outside” the values observed in the dataset (i.e. they won’t be positive integers between 1 and 5)
  - ► In the previous example, we predict at $x = 3.5$ and our prediction is $\hat{y} = 3.439$
  - ► We don’t have a “3.439” in the dataset, but since the response is **ordinal** (i.e. ordered from best (5) to worst (1)), we can still have an intuitive interpretation for numbers like this in the middle

Linear Regression with Categorical Response

- Suppose instead that our response was non-ordinal
  - E.g. Diseases: 1 = “Diabetes”, 2 = “Cancer", ..., 5 = "Cold/Flu"
  - Now we don’t have a nice interpretation for something like “3.439”
- What if there were just not as many classes/categories?

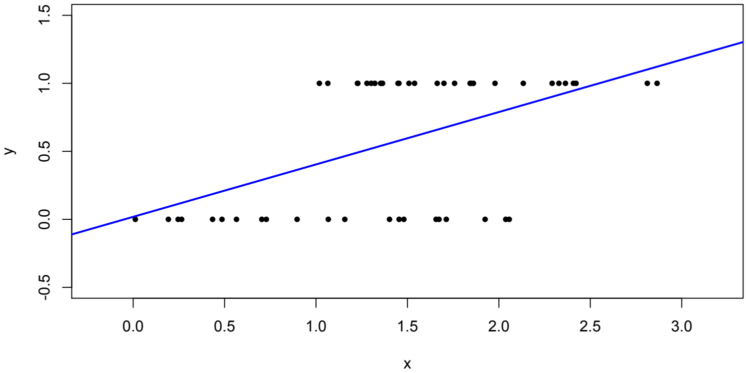
Linear Regression with Binary Response

Now suppose we have a binary response $Y \in \{0, 1\}$ where we take 0 = “Failure" and 1 = “Success"



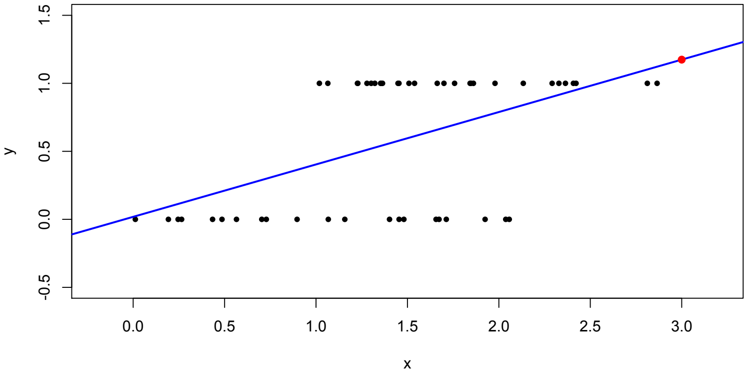
Linear Regression with Binary Response

Now suppose we have a binary response $Y \in \{0, 1\}$ where we take 0 = “Failure" and 1 = “Success"



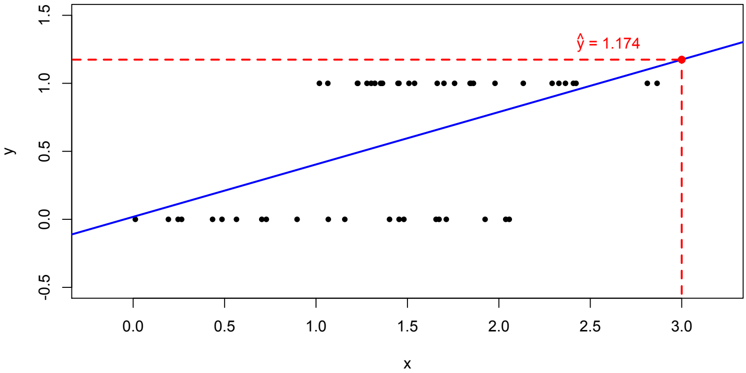
Linear Regression with Binary Response

Now suppose we have a binary response $Y \in \{0, 1\}$ where we take 0 = “Failure" and 1 = “Success"



Linear Regression with Binary Response

Now suppose we have a binary response $Y \in \{0, 1\}$ where we take 0 = “Failure" and 1 = “Success"



Linear Regression with Binary Response

- Now when we fit a linear model, not only are our predictions are almost always going to be something other than exactly “0" or “1", but they may often be outside $[0, 1]$
  - In the previous example, we predict at $x = 3$ and our prediction is $\hat{y} = 1.174$
  - With binary data treated as “Success" (1) and “Failure" (0), we can interpret anything in $[0, 1]$ as the “probability of success”, but things outside that range don’t make sense

Takeaway

- So is it *ever* ok to do linear regression with a categorical response?
  - Yes, kind of, but should always be done with caution
  - As a general rule, you need **(1) ordinal** data (to interpret the “in-between" predictions) and **(2) many classes/categories** (so that at least many of the predictions fall within the category ranges and can be made sense of)
- In many situations though, we either want to predict an actual class/category, or at least want our predictions to be restricted to a given range
- Most of the time, some kind of *generalized* linear model would be more appropriate

Classification

Classification Set-up

- For the remainder of this lecture, we'll assume we have ordered pairs of data of the form (x_i, y_i) for $i = 1, \ldots, n$ where $x_i = (x_{1,i}, \ldots, x_{p,i})$ but with

$$y_i \in \{a_1, a_2, \ldots, a_K\} \text{ for } i = 1, \ldots, n$$

- That is, we assume Y is a categorical variable with K classes/categories
- **Note:** Don't be confused by the "ordered pair" language here; each x_i still represents a vector of p predictor variables. We're just calling that vector x_i as convenient shorthand.

Classifiers

- Given data of this form, the most common goal is to construct a **classifier** (or classification function) f:

$$f : \mathcal{X} \rightarrow \{a_1, \dots, a_K\}$$

- That is, a classifier f is simply a function that takes in the predictor/feature/covariate information $X_1, \dots, X_p$ and assigns a predicted class a_j
- Obvious question then: How should the classifier decide which class it should predict?

Bayes Classifier

Bayes Classifier

- Given that our response variable Y belongs to one of m classes $\{a_1, a_2, ..., a_K\}$, define

  $$p_k(X) = Pr\left[Y = a_k \mid X\right]$$

  as the probability that Y belongs to the kth class (a_k), given the feature/covariate information in X

- Then, given a particular feature/covariate vector x for which we want to predict the class, we would like to calculate

  $$p_1(x),\ p_2(x),\ \ldots\ ,\ p_K(x)$$

Bayes Classifier

- So then given

$$p_1(x),\ p_2(x),\ \dots,\ p_K(x)$$

  what's the obvious way to assign/predict a class for x? Well, simply choose the class k with the largest $p_k(x)$. That is, assign x to the class that it has the highest probability of belonging to. This is the **Bayes Classifier**.

e.g. Suppose $Y \in \{0, 1\}$ and for some x, $p_0(x) = 0.25$.

Since there's only two classes, we know $p_1(x) = 1 - p_0(x) = 0.75$ and thus we assign x to class 1 since $p_1(x) > p_0(x)$.

Bayes Classifier

- Note that while we've made the problem a bit more formal, we haven't really gotten any closer to solving the problem.
  - Hard part is figuring out how to estimate those probabilities $p_k(x)$.

- Also note that though the idea of a Bayes classifier seems obvious, it's not always the one you would want to use.
  - **Why?** We'll talk more formally about this later.

Logistic Regression

Logistic Regression

- Again assume we've got ordered pairs of data (x_i, y_i) with each $x_i = (x_{1,i}, \ldots, x_{p,i})$ but now we assume the **response is binary**: $y_i \in \{0, 1\}$
  - If $y_i \in \{a_1, a_2\} \neq \{0, 1\}$, usually it's possible (reasonable) to call one outcome the success (1) and the other the failure (0)
- As in our previous example, note that since we have a binary response, if we know $p_1(x)$ then we know $p_0(x) = 1 - p_1(x)$. Thus, let's simplify notation and write

$$p(x) = p_1(x)$$

- We want a model for $p(x)$ but recall that we don't want to use a linear model:

$$p(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

Logistic Regression

- Again assume we've got ordered pairs of data (x_i, y_i) with each $x_i = (x_{1,i}, \dots, x_{p,i})$ but now we assume the **response is binary**: $y_i \in \{0, 1\}$
  - If $y_i \in \{a_1, a_2\} \neq \{0, 1\}$, usually it's possible (reasonable) to call one outcome the success (1) and the other the failure (0)
- As in our previous example, note that since we have a binary response, if we know $p_1(x)$ then we know $p_0(x) = 1 - p_1(x)$. Thus, let's simplify notation and write

$$p(x) = p_1(x)$$

- We want a model for $p(x)$ but recall that we don't want to use a linear model:

~~$p(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$~~

Logistic Regression

- Instead, since we know that $p(x)$ denotes the *probability* that x should be assigned to class “1", ideally the model should predict/map into $[0, 1]$

Logistic Regression

- Instead, since we know that $p(x)$ denotes the *probability* that x should be assigned to class “1", ideally the model should predict/map into $[0, 1]$

- Most popular choice of model uses the **logistic** function:

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} \tag{1}$$

Logistic Regression

- Instead, since we know that $p(x)$ denotes the *probability* that x should be assigned to class “1", ideally the model should predict/map into $[0, 1]$

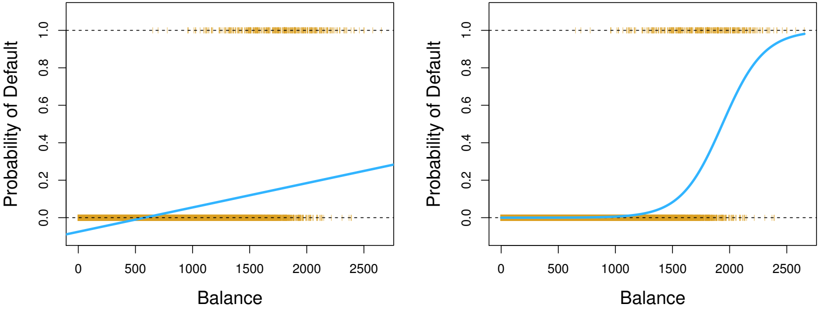
- Most popular choice of model uses the **logistic** function:

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}} \tag{1}$$

- In this case

$$\beta_i > 0 \implies \text{Increase in } X_i \text{ leads to increase in } p(x)$$
$$\beta_i < 0 \implies \text{Increase in } X_i \text{ leads to decrease in } p(x)$$

Logistic Regression



ISL Fig. 4.2: Probability of Default modeled as a function of Balance using Linear Regression (Left) and Logistic Regression (Right).

Logistic Regression

- The logistic regression framework might at first appear a bit complicated and mysterious, but note that with a little algebra:

$$p(x) = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}}$$

$$\iff \quad \frac{p(x)}{1-p(x)} = e^{\beta_0+\beta_1 x_1+\cdots+\beta_p x_p}$$

$$\iff \quad \log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

- The LHS of row 2 is what is often called the **odds** (or odds ratio) and the LHS in the third row is then the **log-odds**

Logistic Regression - Estimating the Parameters

- With linear regression models, recall that we fit the model (i.e. estimated the parameters) via ordinary least squares (OLS):
  - General idea was to define a loss function – squared error in the case of OLS – and choose our coefficient estimates $\hat{\beta}$ to *minimize* that loss
- With logistic regression, we use a more general method: maximum likelihood estimation (MLE)
  - Instead of *minimizing* a loss function, we *maximize* a likelihood function
- **Note:** In linear regression, $\hat{\beta}_{OLS} = \hat{\beta}_{MLE}$ whenever the errors follow a normal distribution

Logistic Regression - Other Important Details

- Much of the same kinds of inference can be carried out with logistic regression as with linear regression
    - z-tests replace the t-tests in linear regression
- Same general procedures in place for dealing with categorical predictors
- To make predictions, estimate $\hat{\beta}_{MLE}$ and plug into (1)
- **Importantly:** Coefficient interpretations are still *with respect to* the other terms in the model
    - Classic examples of where this goes wrong. Book discusses problem with loan defaults; Homework problem on classic Berkeley gender bias case

Other Kinds of Responses

- When the number of response classes is more than 2, $y_i \in \{a_1, \ldots, a_K\}$, we can consider doing *multinomial* logistic regression – two options for this:
  - **Standard/Baseline Coding:** Pick 1 class (outcome) to treat as the default; log odds between any two classes still take the form of a linear model. Choice of baseline only important for interpretation.
  - **Softmax Coding:** Treat all classes symmetrically so that

$$Pr[Y = k|X = x] = \frac{e^{\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p}}{\sum_{l=1}^{K} e^{\beta_{l0}+\beta_{l1}x_1+\cdots+\beta_{lp}x_p}}$$

- In these situations, other tools such as **linear discriminant analysis** and even other kinds of GLMs can also be used as an alternative and are probably more popular

Other Kinds of Responses

- Logistic regression is merely one particular kind of *generalized* linear model (GLM) where we assume

$$\eta(\mathbb{E}(Y|X)) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

  so that the transformed mean is a linear function of X and η is called a link function

- **Linear:** $\eta(\mu) = \mu$ **Logistic:** $\eta(\mu) = log(\mu/(1-\mu))$ **Poisson:** $\eta(\mu) = log(\mu)$
- Many varieties of GLMs out there, many with specialized uses for particular scientific applications. We won't go deeper on this topic in this course.

Linear Discriminant Analysis

Linear Discriminant Analysis

- Again assume we've got ordered pairs of data (x_i, y_i) with each $x_i = (x_{1,i}, ..., x_{p,i})$ but now with $y_i \in \{a_1, ..., a_K\}$ and to simplify notation a bit, let $\{a_1, ..., a_K\} = \{1, 2, .., K\}$ and define:
  - $p_k(x) = Pr\,(Y = k \,|\, x)$ The probability of being in class k given the predictor information x (same as before)
  - $f_k(x) = Pr\,(X = x \,|\, Y = k)$ The distribution of predictors within class k. Note that if this were the same across all k, we wouldn't be able to do much.
  - π_k The probability that a randomly chosen observation (x, y) falls into class k. This does **not** take into account the predictor information in X.

Linear Discriminant Analysis - Bayes Rule

- How can we combine these to get useful information about $p_k(x)$?

Linear Discriminant Analysis - Bayes Rule

- How can we combine these to get useful information about $p_k(x)$?
- **Bayes Rule:** May have seen this in probability written in terms of events (sets):

$$p_k(x) = \frac{\pi_k f_k(x)}{\sum_{i=1}^{K} \pi_i f_i(x)}$$

- **Intuition:** We call the π_k the *prior* probabilities and the $p_k(x)$ the *posterior* probabilities. Bayes Rule is a way to trade-off the information in the prior π_k and the data $f_k(x)$

Bayes Rule Example

Example: Suppose we have 100 data points of the form (x, y) with $y \in \{1, 2\}$ and 80% of that data comes from class 1. Suppose also that we have one categorical predictor $X_1 \in \{A, B\}$ with

$$f_1(A) = Pr(X_1 = A|Y = 1) = 0.6 \quad \text{and} \quad f_2(A) = Pr(X_1 = A|Y = 2) = 0.3.$$

Bayes Rule Example

Example: Suppose we have 100 data points of the form (x, y) with $y \in \{1, 2\}$ and 80% of that data comes from class 1. Suppose also that we have one categorical predictor $X_1 \in \{A, B\}$ with

$$f_1(A) = Pr(X_1 = A|Y = 1) = 0.6 \quad \text{and} \quad f_2(A) = Pr(X_1 = A|Y = 2) = 0.3.$$

Then

$$f_1(B) = Pr(X_1 = B|Y = 1) = 0.4 \quad \text{and} \quad f_2(B) = Pr(X_1 = B|Y = 2) = 0.7$$

Bayes Rule Example

Example: Suppose we have 100 data points of the form (x, y) with $y \in \{1, 2\}$ and 80% of that data comes from class 1. Suppose also that we have one categorical predictor $X_1 \in \{A, B\}$ with

$$f_1(A) = Pr(X_1 = A|Y = 1) = 0.6 \quad \text{and} \quad f_2(A) = Pr(X_1 = A|Y = 2) = 0.3.$$

Then

$$f_1(B) = Pr(X_1 = B|Y = 1) = 0.4 \quad \text{and} \quad f_2(B) = Pr(X_1 = B|Y = 2) = 0.7$$

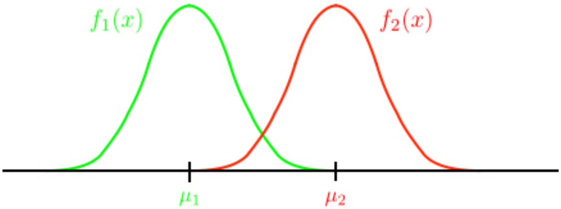
So the probability that y^* belongs to class 1 if we observe $x_1^* = A$ is given by

$$p_1(A) = Pr(Y = 1|X_1 = A) = \frac{\pi_1 f_1(A)}{\sum_{i=1}^{2} \pi_i f_i(A)} = \frac{0.8(0.6)}{0.8(0.6) + 0.2(0.3)} = 0.889$$

Linear Discriminant Analysis Set-up

- Assume that we have only a single predictor X with $Y \in \{1, 2\}$ and that $f_k(x)$ has a normal distribution for both $k = 1, 2$ with the *same* variance and different means:

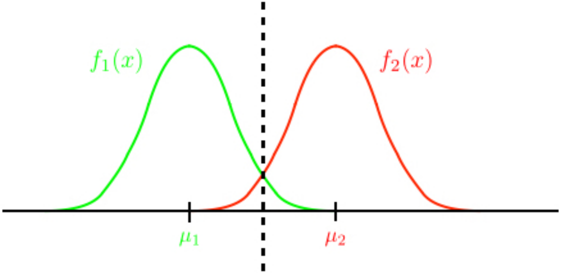
$$f_1(x) = \mathcal{N}(\mu_1, \sigma^2) \qquad f_2(x) = \mathcal{N}(\mu_2, \sigma^2)$$



- If $y = 1$, we assume x is a sample from $f_1(x)$; If $y = 2$, we assume x is a sample from $f_2(x)$

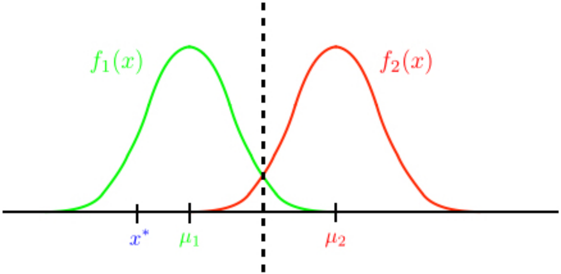
Linear Discriminant Analysis

- Question is: Given some particular observation x^*, how do I decide which distribution it most likely came from? **Need to establish some decision boundary**



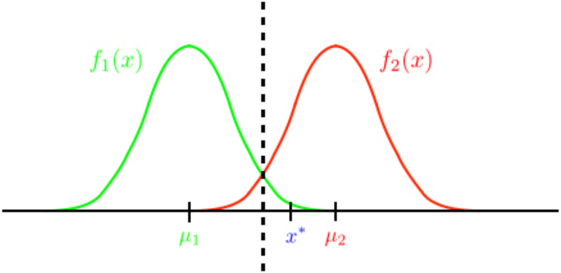
Linear Discriminant Analysis

- Question is: Given some particular observation x^*, how do I decide which distribution it most likely came from? **Need to establish some decision boundary**



Linear Discriminant Analysis

- Question is: Given some particular observation x^*, how do I decide which distribution it most likely came from? **Need to establish some decision boundary**



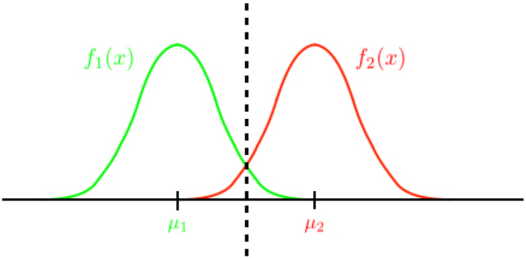
LDA Boundaries

- **Very Important:** Note that if $\pi_1 = \pi_2 = 0.5$, then this decision boundary would simply be at the midpoint of the two normal distributions. **However**, if, for example, $\pi_1 > \pi_2$, then the decision boundary shifts so as to make it more likely that y would be assigned to class 1.

- **Why? What's the intuition?** Well, if $\pi_1 > \pi_2$, that means our *prior* belief – independent of X – is that in general, class 1 is more likely than class 2. Thus, in situations where it's "close", we would "err" on the side of choosing class 1.

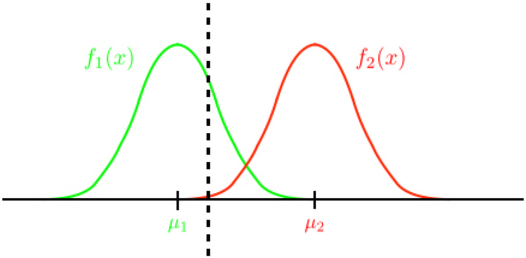
LDA Boundaries

$$\pi_1 = \pi_2$$



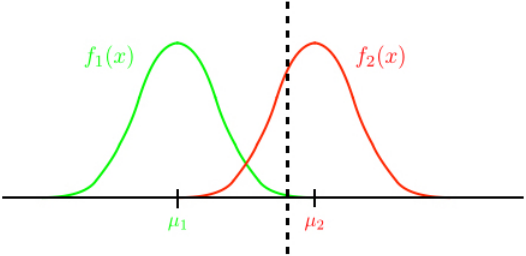
LDA Boundaries

$\pi_1 < \pi_2$



LDA Boundaries

$$\pi_1 > \pi_2$$



LDA in Practice

- In practice, we have n observations

  $$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

  from which we need to estimate the means (locations) of the two distributions, as well as π_1 and π_2

- We can follow the same general procedure discussed in ISLR Chapter 2: split data in *train* and *test* sets, estimate $f_i(x)$, π_i with training set and evaluate the performance on the test set
  - These can then be plugged into Bayes rule to find the appropriate boundary

Kinds of Errors

- There are two kinds of errors we can make with classification problems: predict $\hat{y} = 1$ when $y = 2$ or predict $\hat{y} = 2$ when $y = 1$
  - ► Goes for *any* binary classification problem; not specific to LDA
  - ► We can summarize results in a **confusion matrix:**

	Truth			
		No	**Yes**	**Total**
Predicted	**No**	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

Table: (ISLR Table 4.4) Confusion matrix from predicting credit defaults.

Kinds of Errors

- **Note:** Why not just report the overall misclassification rate?

$$\frac{1}{n}\sum_{i=1}^{n} 1\{\hat{y}_i \neq y_i\} = \frac{\text{\# misclassifications}}{\text{\# total observations}}$$

Kinds of Errors

- **Note:** Why not just report the overall misclassification rate?

$$\frac{1}{n}\sum_{i=1}^{n} 1\{\hat{y}_i \neq y_i\} = \frac{\text{\# misclassifications}}{\text{\# total observations}}$$

- Because there may be a higher cost (bigger loss) in wrongly predicting one class or another
  - E.g. Suppose two classes are “Cancerous” vs. “Not Cancerous” – if decision is whether to take a biopsy, we’d much rather err on the side of predicting cancer (taking a biopsy)

Evaluating Classifier Quality

- In most binary classification settings, it’s natural to define one outcome as the success/positive and the other as the failure/negative
  - E.g. $Y \in \{$Has Disease, Doesn’t Have Disease$\}$, “Has Disease" typically the positive outcome
- In this context, we define

Sensitivity: (True Positive Rate) - Proportion of positive samples correctly classified

Specificity: (True Negative Rate) - Proportion of negative samples correctly classified

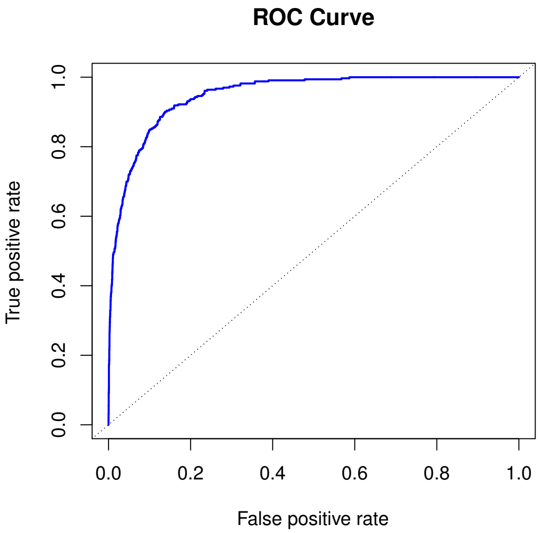
Evaluating Classifier Quality

- Because in many scenarios there are different costs/losses for making different kinds of mistakes, the Bayes Classifier (assigns predicted class as that with the highest predicted probability) is often not always the best choice

- More often, we choose some *threshold* t so that, for example, we predict class “0" unless $p_1(x) > t$ (and where t needn’t be equal to 0.5)

- An **ROC** (receiver operating characteristic) curve summarizes the performance of a binary classifier in terms of sensitivity and specificity across a range of thresholds

ROC Curves



ISLR Figure 4.8: An ROC Curve

ROC Curves

- The **A**rea **U**nder the ROC **C**urve is called the AUC
- Larger AUC values are better, so ideally the ROC will hug the top left corner of $[0,1] \times [0,1]$
- A classifier that randomly assigns classes would have an expected AUC of 0.5 (diagonal ROC curve), so we'd never expect a classifier to be below the diagonal. **Why?**
- So how do you choose a threshold in practice? Depends on what kind of false positive and/or false negative rates you want. (Gets back to idea of different losses for different mistakes.)

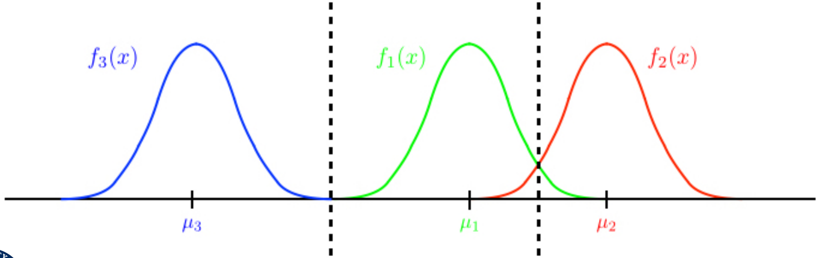
LDA Extensions

LDA for > 2 classes

- Thus far we've covered two procedures:
  - Logistic Regression (2 classes (usually), p predictors)
  - LDA (2 classes, 1 predictor)
- If we have one predictor but more than 2 classes, we can just add additional decision boundaries:

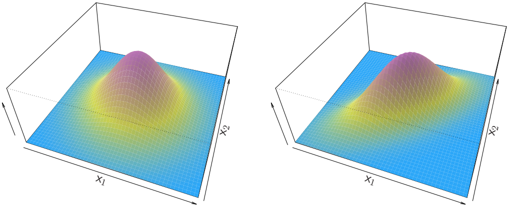
LDA for > 2 classes

- Thus far we've covered two procedures:
  - Logistic Regression (2 classes (usually), p predictors)
  - LDA (2 classes, 1 predictor)
- If we have one predictor but more than 2 classes, we can just add additional decision boundaries:



LDA for > 1 predictor

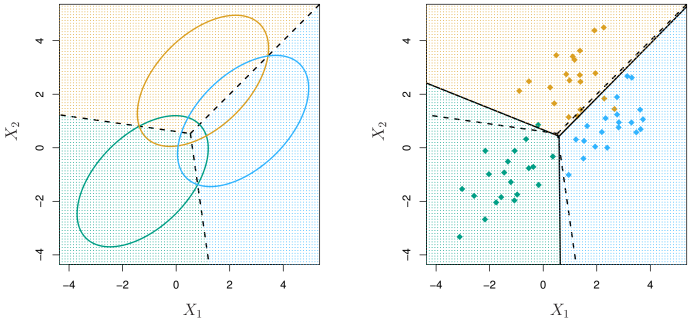
- If we have more than 1 predictor, we can carry out the same kind of procedure, but now we're dealing with multivariate normal distributions – that is, we assume that observations (predictors) from the k^{th} class come from $\mathcal{N}(\mu_k, \mathbf{\Sigma})$ where μ_k is a mean vector and $\mathbf{\Sigma}$ is a common covariance matrix



ISLR Figure 4.5: Multivariate Normal Distributions

LDA with many predictors and classes

- With > 1 predictor and > 2 (response) classes, the decision boundaries look something like:



ISLR Figure 4.6: LDA with 2 predictors and 3 (response) classes. Left: Ellipses corresponding to (true) 95% probabilities for each class with (true) Bayes decision boundary as dashed line. Right: 20 points sampled from each class and LDA decision boundaries as solid lines.

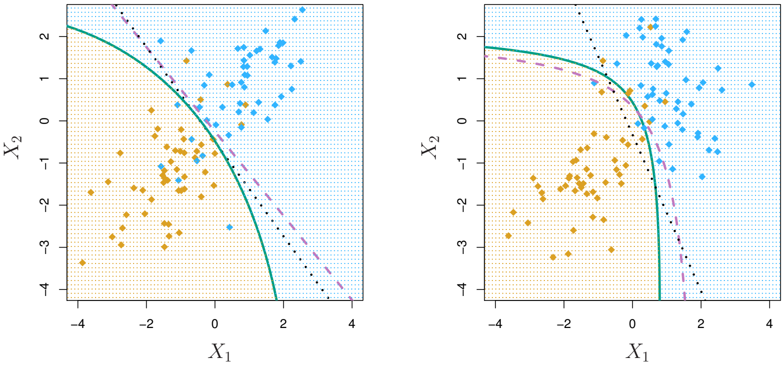
Quadratic Discriminant Analysis

Quadratic Discriminant Analysis

- Recall that in LDA, we assumed the class-conditional predictor distributions were Normally distributed with the *same variance*: $f_i(x) = \mathcal{N}(\mu_i, \sigma^2)$ or $f_i(x) = \mathcal{N}(\mu_i, \boldsymbol{\Sigma})$ in the multivariate case (multiple predictors)

- **Quadratic Discriminant Analysis** (QDA) is the natural extension of this: we assume each $f_i(x)$ has its *own* variance (or covariance matrix if $p > 1$)
  - This creates a **quadratic** decision boundary as opposed to the **linear** decision boundary created by LDA

Quadratic Discriminant Analysis



ISLR Figure 4.9: Two different binary classification problems:

Purple Dashed Line = (True) Bayes decision boundary

Green Solid Line = QDA Decision Boundary

Black dotted line = LDA Decision Boundary.

Note: On the left, $\Sigma_1 = \Sigma_2$. On the right, $\Sigma_1 \neq \Sigma_2$.

Quadratic Discriminant Analysis

- What does this mean in terms of flexibility?

Quadratic Discriminant Analysis

- What does this mean in terms of flexibility?
  - QDA is more flexible – less bias, higher variance relative to LDA

Quadratic Discriminant Analysis

- What does this mean in terms of flexibility?
  - QDA is more flexible – less bias, higher variance relative to LDA

- Which (LDA or QDA) would be more preferred with large datasets (i.e. when n is large relative to p)?

Quadratic Discriminant Analysis

- What does this mean in terms of flexibility?
    - QDA is more flexible – less bias, higher variance relative to LDA

- Which (LDA or QDA) would be more preferred with large datasets (i.e. when n is large relative to p)?
    - Since QDA is more flexible, it can adapt to more kinds of (true) decision boundaries, even if they're linear. Even though we estimate multiple variances/covariances with QDA, those estimates will eventually be very similar given enough data (large n).

Quadratic Discriminant Analysis

- Let's examine this from a slightly different perspective though. Keep in mind that with multiple predictors, our covariance matrix takes the form:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,p} \\ \sigma_{2,1} & \sigma_2^2 & \cdots & \sigma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p,1} & \sigma_{p,2} & \cdots & \sigma_p^2 \end{pmatrix}$$

  where $\sigma_i^2 = \text{Var}(X_i)$ and $\sigma_{i,j} = \sigma_{j,i} = \text{Cov}(X_i, X_j)$.

- This is a diagonal matrix, but still requires the estimation of $p(p+1)/2$ parameters

Quadratic Discriminant Analysis

How does this affect performance?

- If p is large, then $p(p+1)/2$ parameters is a lot to estimate
- If there are K possible classes for Y and we do QDA, then we need to estimate $Kp(p+1)/2$ parameters $\implies$ that's *really* a lot if p is large
- In the same way that adding terms increases variance in linear models (potential for overfitting), adding more variance parameters to estimate does the same in this setting
- Importantly: This means that the “true" underlying boundary is not the only consideration – if we have relatively little data, LDA may produce a better fit even if the boundary is non-linear because it has lower variance and higher bias

Naive Bayes

Naive Bayes

- Recall that in order to apply Bayes rule, we need to provide/estimate class-conditional distributions for the predictors $f_i(X)$
- In LDA (and QDA) we assume that those distributions are normal with the same (or different) variance/covariance matrix
- Naive Bayes takes a much different approach: the defining property/assumption of Naive Bayes is that within each of the K classes, the predictors are independent

  $$\implies f_i(x) = f_{i,1}(x) \cdot f_{i,2}(x) \cdots f_{i,p}(x)$$

  The joint distribution of X can be written as the product of the marginals

Naive Bayes

- Marginal distributions can be estimated however we like – nonparametric density estimation is one choice but this incurs extra variance over parametric approaches
- Alternatively, we could use a normal distribution for each (continuous) feature. Similar to QDA, but independence between features means that off-diagonal elements of Σ are equal to 0 $\implies$ fewer parameters to estimate so less variance in the procedure

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,p} \\ \sigma_{2,1} & \sigma_2^2 & \cdots & \sigma_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p,1} & \sigma_{p,2} & \cdots & \sigma_p^2 \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \sigma_p^2 \end{pmatrix}$$

Naive Bayes

- Is that independence assumption likely to hold in real-world settings?
- Almost certainly not – so why do this?
- Again this comes back to a bias-variance tradeoff – independence assumption gives a big variance reduction. In practice, it’s possible the benefits of that variance reduction outweigh the costs of not modeling dependencies.

k-Nearest Neighbors

k-Nearest Neighbors

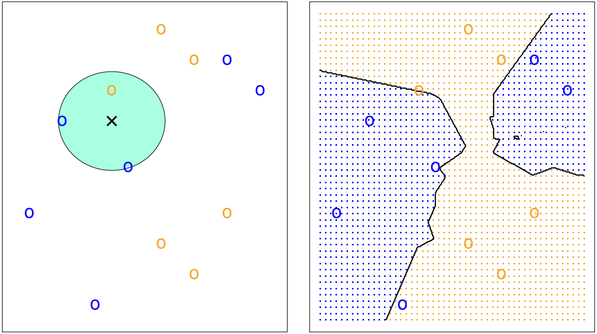
- The final classification method we'll talk about in this chapter is k-Nearest Neighbors (kNN). This method stands in stark contrast to all others we've looked at thus far:

- Logistic Regression, LDA, QDA, Naive Bayes, and even Linear Regression are **global methods**:
  - Every point in the (training) dataset has some influence on the estimated model and therefore on the prediction

- kNN is a **local method**: the prediction at a given location x^* depends only on the points in the (training) dataset closest to x^*

kNN Overview

- Given a particular point x^* where we want to make a prediction, find the k closest points to x^* and take a **majority vote** amongst those k points – assign $\hat{y}^*$ to most frequent class.



ISLR Figure 2.14: k-nearest neighbors with $k = 3$. Resulting decision boundary is shown on the right.

k-Nearest Neighbors

- What do you notice immediately looking at the decision boundary? How is it different from LDA/QDA?

k-Nearest Neighbors

- What do you notice immediately looking at the decision boundary? How is it different from LDA/QDA?
  - *Extremely* flexible. Not only are the boundaries not restricted to be linear or quadratic, there may be many of them

- Formally, what is kNN really doing?

k-Nearest Neighbors

- What do you notice immediately looking at the decision boundary? How is it different from LDA/QDA?
  - *Extremely* flexible. Not only are the boundaries not restricted to be linear or quadratic, there may be many of them

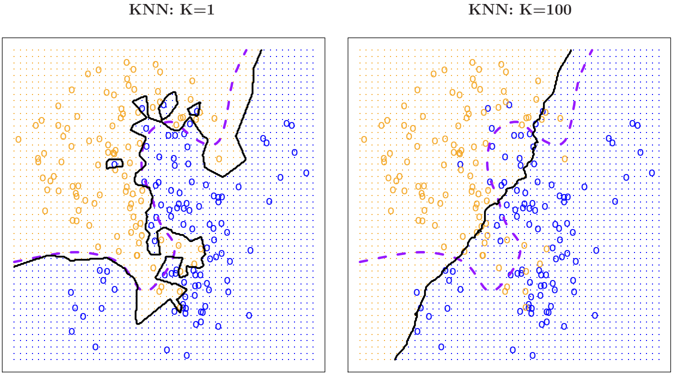
- Formally, what is kNN really doing? We're estimating

$$p_i(x) = Pr(Y = i|X)$$

just like all of the other methods, but in a completely non-parametric way. We're not assuming *any* models or assumptions on the data – simply going off the data that's closest by.

kNN Overview

The amount of flexibility of kNN depends entirely on the choice of k:


KNN: K=1

KNN: K=100

ISLR Figure 2.16: kNN decision boundaries in black; (true) Bayes decision boundaries in purple. (Left) k-nearest neighbors with $k = 1$ – very flexible. (Right) k-nearest neighbors with $k = 100$ – *much less* flexible.

kNN Regression

- Here we defined kNN for classification, but the same idea can be applied in a regression context in which $Y \in \mathbb{R}$:
  - Exact same general process: given a point x^* where we want to make a prediction, start by finding the k closest points. To get an estimate $\hat{y}^*$, simply take the average of those k points:

$$\hat{y}^* = \frac{1}{k} \sum_{i=1}^{n} x_i \, 1\{x_i \in K_{x^*}\}$$

    where K_{x^*} denotes the set of k closest points to x^*, so that $1\{x_i \in K_{x^*}\} = 1$ if x_i is one of the k closest points and otherwise is equal to 0.

k-Nearest Neighbors

- **Big obvious question still remaining:**

k-Nearest Neighbors

- **Big obvious question still remaining:** How should we choose k?

k-Nearest Neighbors

- **Big obvious question still remaining:** How should we choose k?
- We can still make train-test split on data to check how well we're doing, but this only gives us a single estimate of the error.
- More generally, how can I confidently choose between all of these models: Logistic Regression, LDA, QDA, kNN for several choices of k?
  - This takes us to the subject we'll start with in Lecture 5 – Cross-Validation

Lecture 4 Summary

- We've looked at 5 different classification methods: Logistic Regression, LDA, QDA, Naive Bayes, and kNN
- Logistic Regression, LDA, and QDA fall under the heading of parametric statistical methods – Naive Bayes may as well depending on how we choose to model the marginal densities
- k-nearest neighbors is a non-parametric method (could also count as our first machine learning method). Naive Bayes is also something that would appear in most machine learning textbooks.

Lecture 4 Summary

How do they compare in terms of flexibility?

- Most importantly: none of these methods is universally better than any other
- Logistic Regression and LDA generally considered the most rigid – high bias and low variance
- QDA also somewhat rigid, but higher variance due to estimating a new covariance matrix for each class (especially when number of classes and/or predictors is large)
- Naive Bayes and kNN probably have the widest “range of flexibility" – kNN very flexible for small k and very rigid for large k (consider $k = n$, for example). Naive Bayes can be quite flexible depending on how marginals are estimated

Lecture 4 Summary

How do they relate to each other?

- LDA and logistic regression both assume that the log odds are linear in X. QDA assume log odds are quadratic in X. With Naive Bayes, log odds are additive in X (more on generalized additive models later in the course)
- Obviously, LDA is a special case of QDA where the same Σ is assumed for each class
- Not obviously, LDA is actually a special case of Naive Bayes (true for any classifier with a linear decision boundary)
- Naive Bayes with Normal class conditional distributions is the same as LDA with Σ restricted to a diagonal matrix
- Naive Bayes has potential for more flexibility than QDA – QDA might be preferable when there are interactions between predictors

Lecture 4 Summary

Final Big Disclaimer:

- Book gives several empirical comparisons at the end of Chapter 4 and sometimes suggests that some methods do better in settings where their respective assumptions are met
- This may sometimes be true but don't forget about bias and variance – methods that violate assumptions but reduce variance might be preferable in some settings
- Important related issue not mentioned in the book: how *noisy* is the data? That is, how big is the variance of ϵ relative to that of $f(X)$? If it's large, "incorrect" models with low variance may be preferable to "correctly specified" flexible models even if the true f is very jumpy/flexible – more on this topic later in the course