Classification

Szu-Chi Chung

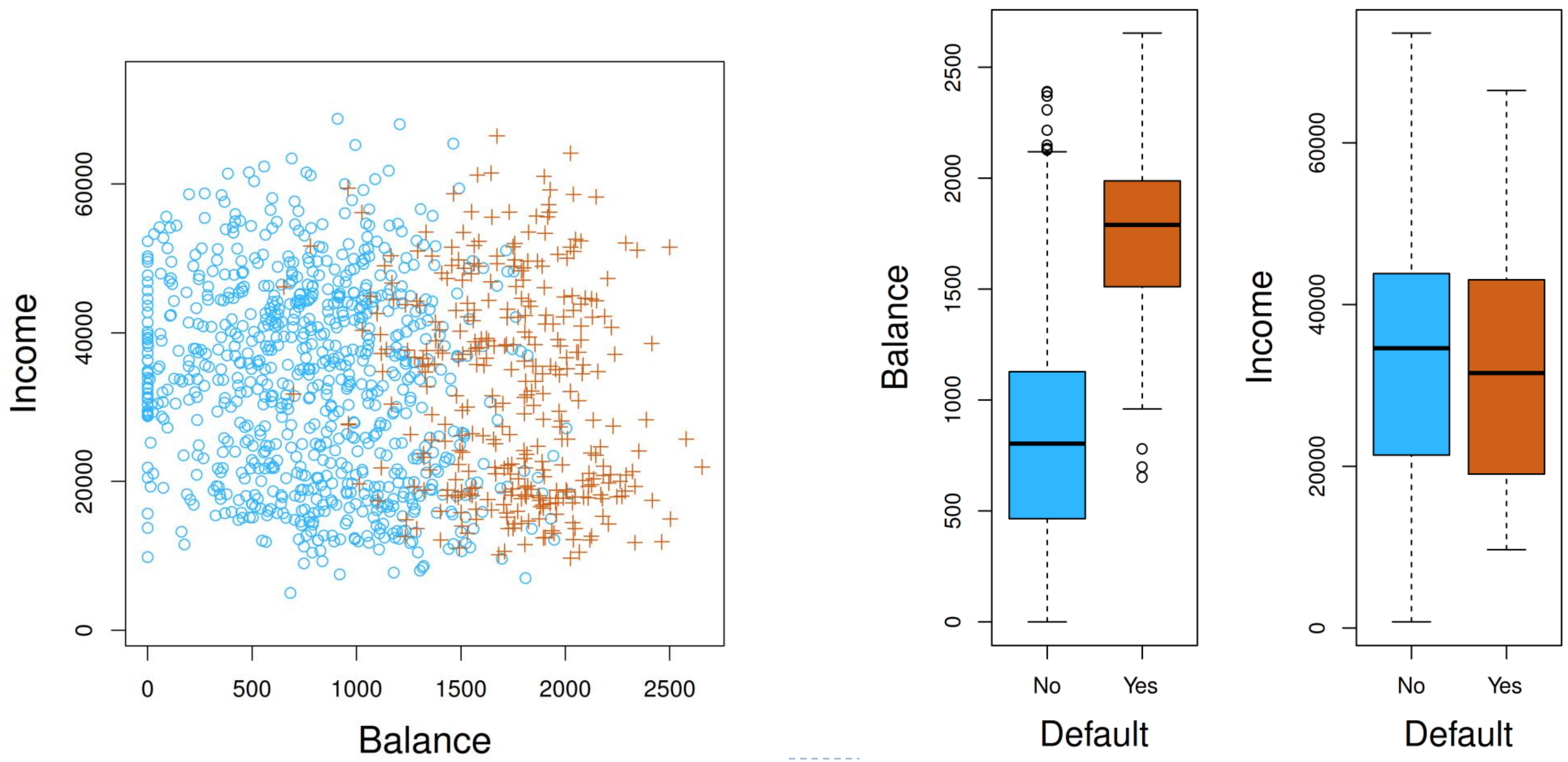
Department of Applied Mathematics, National Sun Yat-sen University

Classification

- Qualitative variables take values in an unordered set C, such as:
  - Eye color ∈ {brown, blue, green}
  - Email ∈ {spam, ham}
- Given a feature vector X and a qualitative response Y taking values in the set C, the classification task is to build a function $C(X)$ that takes as input the feature vector X and predicts its value for Y; i.e. $C(X) \in C$
- Often we are more interested in estimating the *probabilities* that X belongs to each category in C
  - For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent, than a classification fraudulent or not

Example Dataset: Credit Card Default

- $n = 10{,}000, p = 4$ and the default rate is 3.3%



Can we use Linear Regression?

- Suppose we have a response variable with three possible values. We must classify patients according to their symptoms. We can have the code

$$Y = \begin{cases} 1 & \text{if stroke;} \\ 2 & \text{if drug overdose;} \\ 3 & \text{if epileptic seizure.} \end{cases}$$

- This coding suggests an *ordering*, and in fact implies that the *difference* between stroke and drug overdose is the same as between drug overdose and epileptic seizure

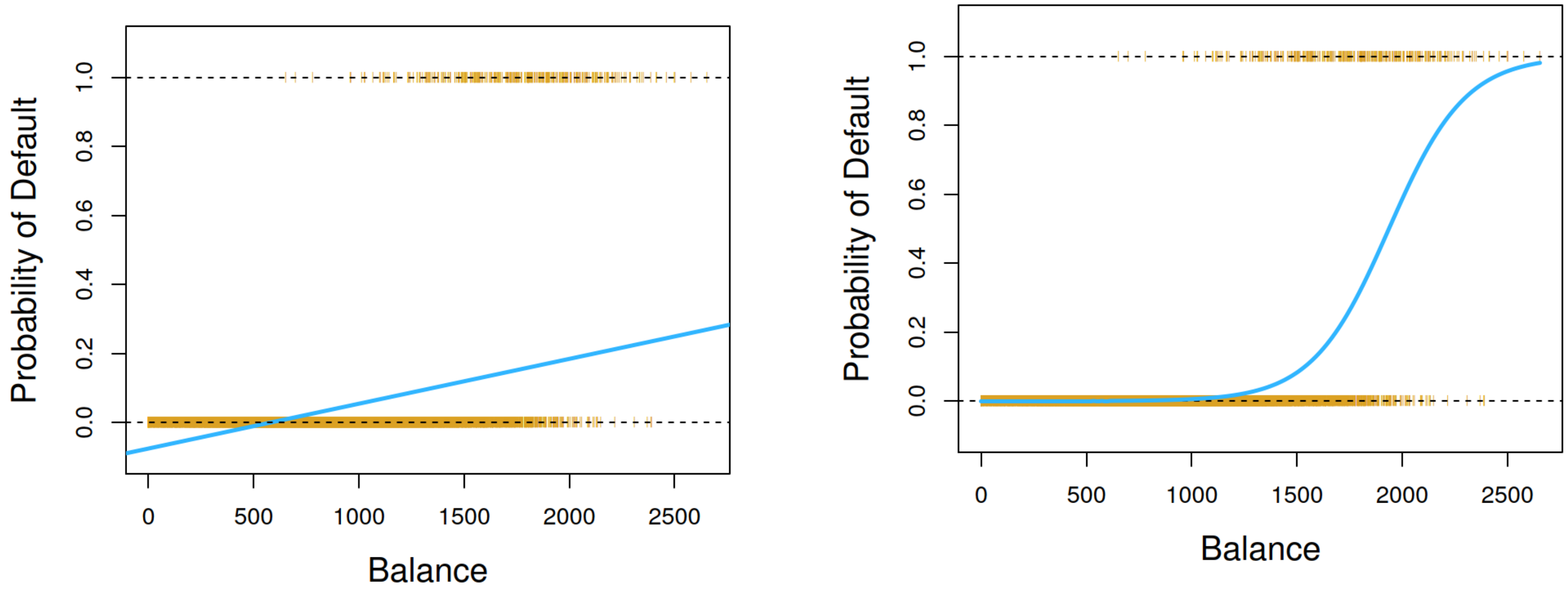
Can we use Linear Regression?

- For the binary response in the Default classification task that we code

$$Y = \begin{cases} 1 & \text{if Yes} \\ 0 & \text{if No} \end{cases}$$

- Can we perform a linear regression of Y on X and classify as Yes if $\hat{Y} > 0.5$?
  - In the binary case it is not hard to show that even if we flip the above coding, linear regression will produce the same final predictions
  - In this case of a binary outcome, linear regression does a good job as a classifier, and is *related* to linear discriminant analysis which we discuss later (ESL exercise 4.2)

Can we use Linear Regression?



- The orange marks indicate the response Y, either 0 or 1. Linear regression does not estimate Pr(Y = 1|X) well. Linear regression might produce probabilities less than zero or bigger than one
  - Logistic regression seems well suited!

1. Logistic Regression

- Let’s write $p(X) = Pr(Y = 1|X)$ for short and consider using `balance` to predict default. Logistic regression uses the form

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

$$Y|X = \text{Bernoulli}(p(X))$$

(e ≈ 2.71828 is a mathematical constant [Euler’s number])

  - No matter what values β_0, β_1 or X take, $p(X)$ will have values between 0 and 1

- A bit of rearrangement gives

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X$$

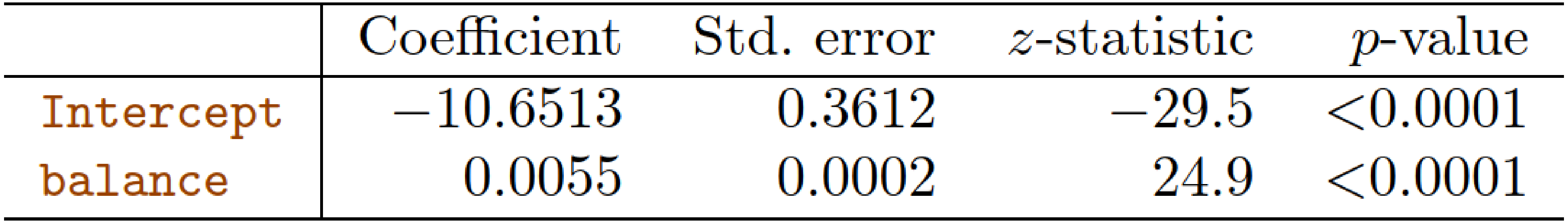
  - This monotone transformation is called the log odds or logit transformation of $p(X)$
  - Note that the decision boundary is still linear

Estimating the Regression Coefficients - Maximum Likelihood

- We use maximum likelihood to estimate the parameters

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} 1 - p(x_i)$$

- This likelihood gives the probability of the observed zeros and ones in the data. We pick β_0 and β_1 to maximize the likelihood of the observed data
- Most statistical packages can fit linear logistic regression models by maximum likelihood (z-statistics or t-statistics?)



	Coefficient	Std. error	z-statistic	p-value
Intercept	-10.6513	0.3612	-29.5	<0.0001
balance	0.0055	0.0002	24.9	<0.0001

Making Predictions

- What is our estimated probability of `default` for someone with a balance of $1000?

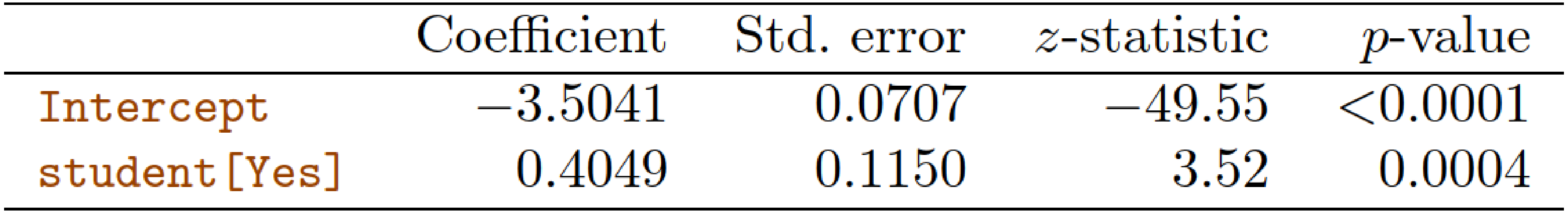
$$\hat{P}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

- With a balance of $2000?

$$\hat{P}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

Making Predictions

- Lets do it again, using `student` as the predictor



	Coefficient	Std. error	z-statistic	p-value
`Intercept`	−3.5041	0.0707	−49.55	<0.0001
`student[Yes]`	0.4049	0.1150	3.52	0.0004

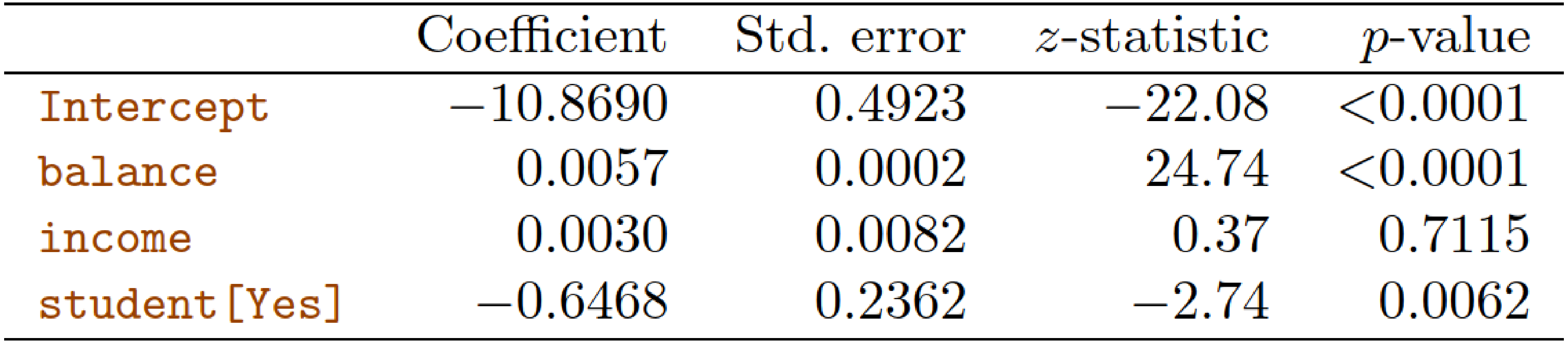
$$\widehat{Pr}(default = Yes|student = Yes) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431$$

$$\widehat{Pr}(default = Yes|student = No) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292$$

Logistic regression with several variables

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}$$

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

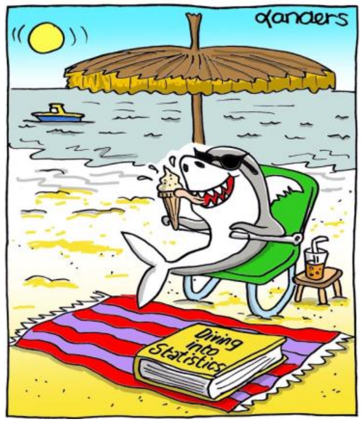


	Coefficient	Std. error	z-statistic	p-value
`Intercept`	-10.8690	0.4923	-22.08	<0.0001
`balance`	0.0057	0.0002	24.74	<0.0001
`income`	0.0030	0.0082	0.37	0.7115
`student[Yes]`	-0.6468	0.2362	-2.74	0.0062

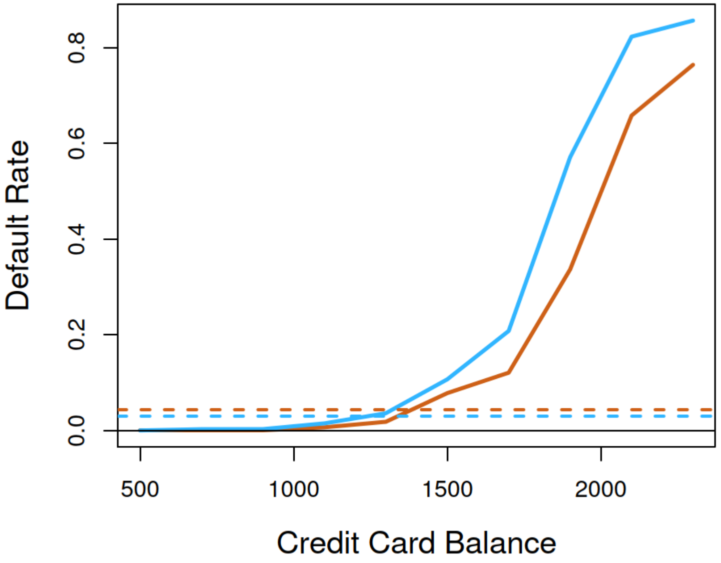
Why is coefficient for student negative, while it was positive before?

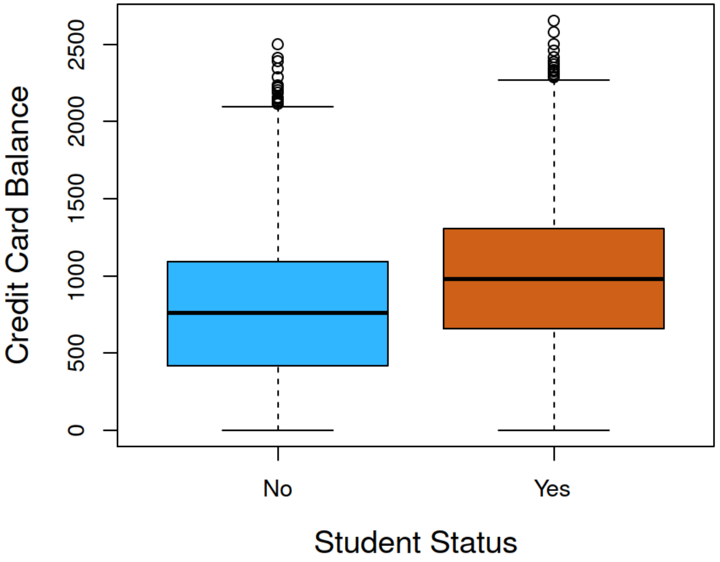
Confounding

- If we have the information about balance and income, then for each level of balance, students default less in multiple logistic regression
  - Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students!
- Multiple logistic regression can tease this out



https://www.causeweb.org/cause/resources/fun/cartoons/ice-cream-sales-and-shark-sightings





Multinomial logistic regression - with more than two classes

- So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes:

$$Y|X = \text{Categorical }(p(X))$$

$$Pr(Y = k|X) = \frac{e^{\beta_{k0}+\beta_{k1}X_1+\beta_{k2}X_2+\cdots+\beta_{kp}X_p}}{\sum_{l=1}^{K} e^{\beta_{l0}+\beta_{l1}X_1+\beta_{l2}X_2+\cdots+\beta_{lp}X_p}}$$

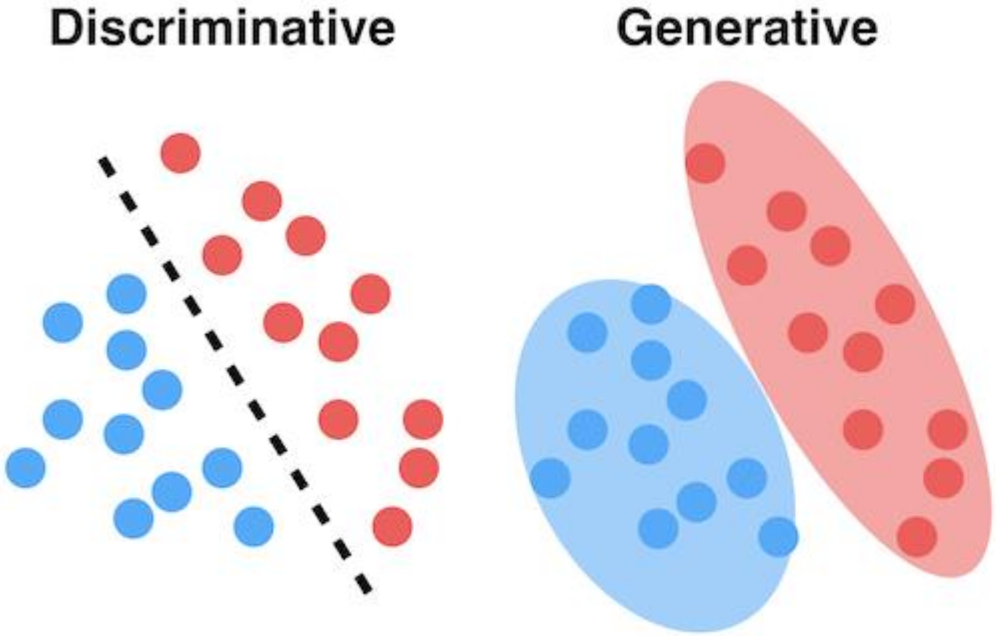
$$\log\left(\frac{\Pr(Y = k|X = x)}{\Pr(Y = k'|X = x)}\right) = (\beta_{k0}-\beta_{k'0}) + (\beta_{k1}-\beta_{k'1})X_1 + \cdots + (\beta_{kp}-\beta_{k'p})X_p$$

- This is similar to the *softmax* activation function used in the neural network model
- Here, we actually estimate coefficients for all K classes
  - Multinomial logistic regression is also referred to as multiclass logistic regression

2. Why use the other approaches?

1. When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem
2. If n is small and the distribution of the predictors X is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model
3. Linear discriminant analysis is popular when we have more than two response classes and it also provides *low-dimensional views* of the data

Generative Models for Classification



https://www.analyticsvidhya.com/blog/2021/07/deep-understanding-of-discriminative-and-generative-models-in-machine-learning/

- Model the distribution of X in each of the classes separately, and then use Bayes theorem to flip things around and obtain $\Pr(Y \mid X)$
  - When we use normal (Gaussian) distributions for each class, this leads to linear or quadratic discriminant analysis. Other distributions can be used as well!

Bayes theorem for classification

- According to the Bayes' theorem:

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

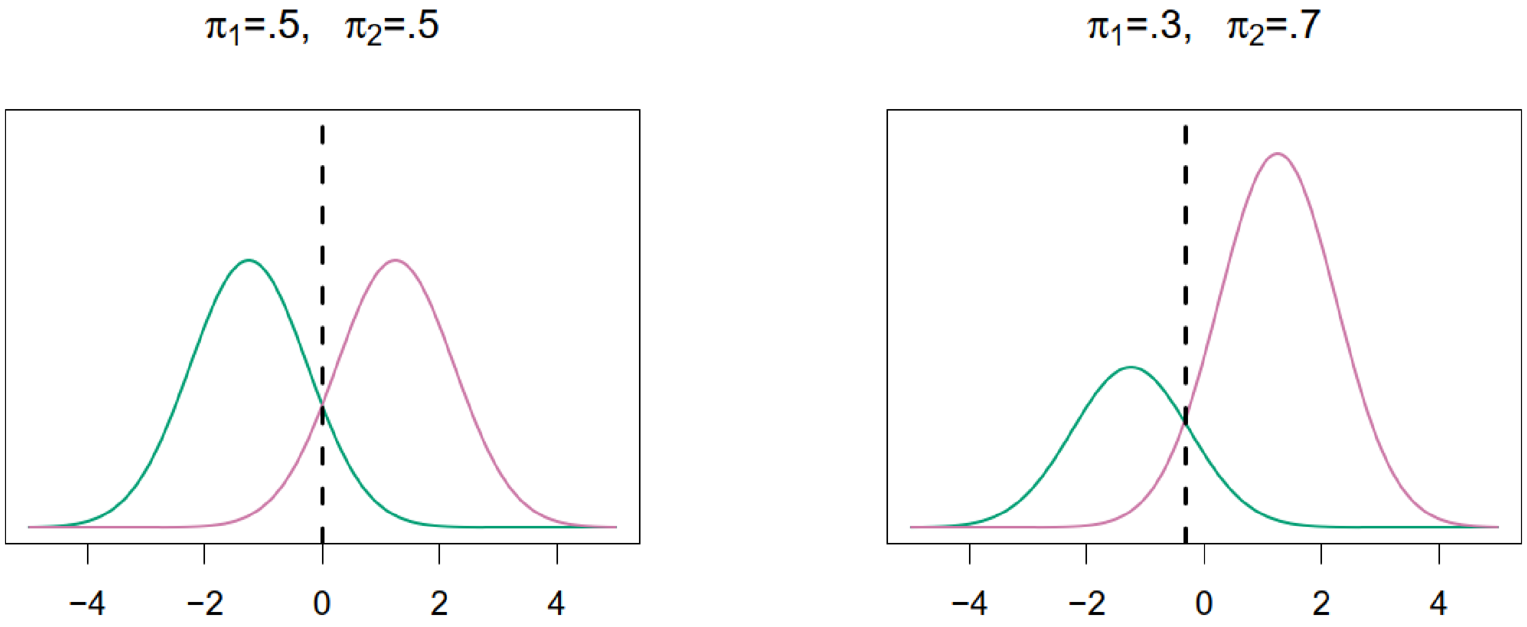
One writes this for discriminant analysis:

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

  - $f_k(x) = Pr(X = x|Y = k)$ is the density for X in class k. Here we will use normal densities for these, separately in each class
  - $\pi_k = Pr(Y = k)$ is the marginal or prior probability for class k

- We discuss three classifiers that use different estimates of $f_k(x)$ to approximate the Bayes classifier: *linear discriminant analysis, quadratic discriminant analysis, and naive Bayes*

Classify to the highest density



- We classify a new point according to which density is highest
  - When the priors are different, we take them into account as well, and compare $\pi_K f_k(x)$. On the right, we favor the pink class — the decision boundary has shifted to the left
  - $Pr(Y = k|X = x)$ is the posterior probability

Linear Discriminant Analysis when $p = 1$

- The Gaussian density has the form

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

  - Here μ_k is the mean, and σ_k^2 the variance (in class k). We will assume that all the $\sigma_k = \sigma$ are the same

- Plugging this into Bayes' formula, we get a rather complex expression for $p_k(x) = Pr(Y = k|X = x)$:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}(\frac{x-\mu_k}{\sigma})^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}(\frac{x-\mu_l}{\sigma})^2}}$$

Estimating the parameters – Maximum likelihood

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

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

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

- Where $\hat{\sigma}_k^2 = \frac{1}{n_k - 1} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$ is the usual formula for the estimated variance in the kth class and n_k is the training sample in the kth class
- We normalize by the scalar $n - K$. When we fit a maximum likelihood estimator it should be divided by n, but if it is divided by $n - K$, we get an unbiased estimator

Discriminant functions

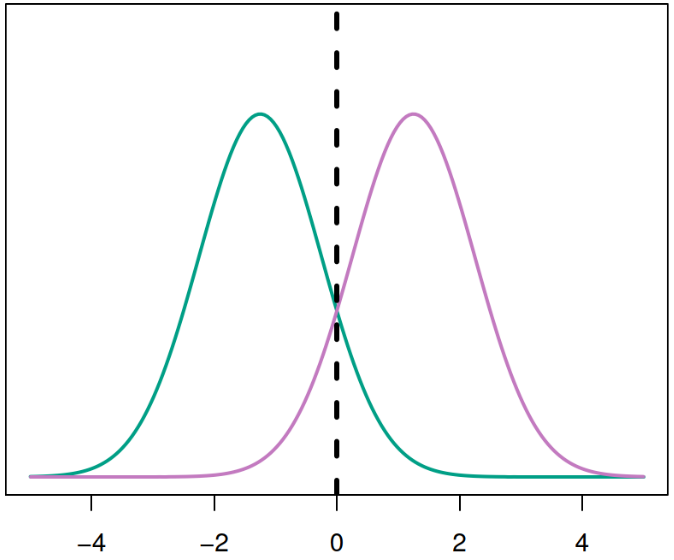
- To classify at the value $X = x$, we need to see which of the $p_k(x)$ is largest. Taking logs of $p_k(x)$, and discarding terms that do not depend on k, this is equivalent to assigning x to the class with the largest discriminant score:

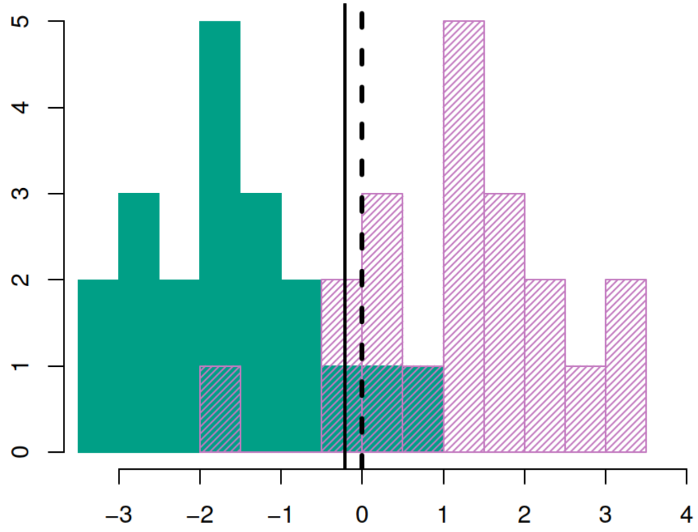
$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k) \text{ (Exercise 4.2)}$$

  - The above is called the discriminant function and note that $\delta_k(x)$ is a linear function of x and when $argmax_k p_k(x) = argmax_k \delta_k(x)$ it results in linear decision boundary
  - If there are $K = 2$ classes and $\pi_1 = \pi_2 = 0.5$, then one can see that the decision boundary is at

$$x = \frac{\mu_1 + \mu_2}{2}$$

Discriminant functions – simulation data





- Example with $\mu_1 = -1.25,\ \mu_2 = 1.25, \pi_1 = \pi_2 = 0.5$, and $\sigma^2 = 1$.
- Typically we don't know these parameters; we just have the training data. In that case we simply estimate the parameters and plug them into the rule

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

Linear Discriminant Analysis when $p > 1$

- Density: $f_k(x) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}} e^{\frac{-1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)}$

Assuming the same covariance matrix

- Discriminant function: $\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log(\pi_k)$

$$= \frac{-1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \log(\pi_k) + C$$

Mahalanobis distance between x and μ_k

- Despite its complicated form

$\delta_k(x) = c_{k0} + c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kp}x_p$ is a linear function

The decision boundary $\{x: \delta_k(x) = \delta_l(x)\}, 1 \le k, l \le K$ is also a linear function

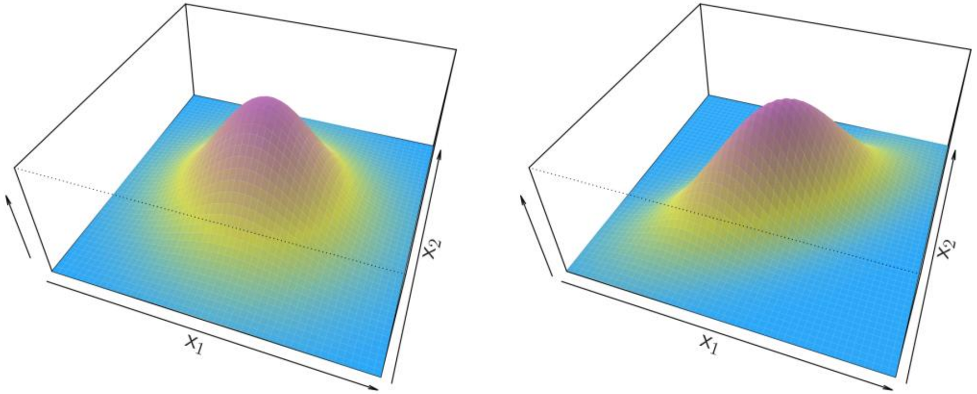
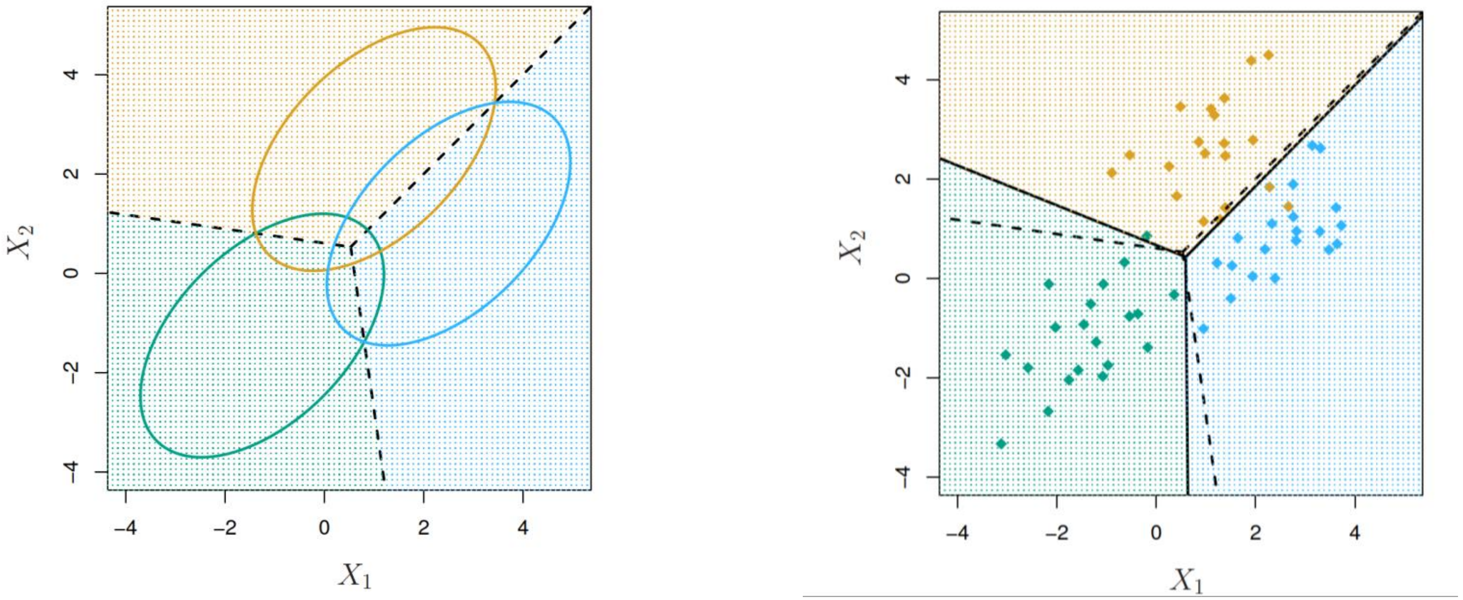


Illustration: $p = 2$ and $K = 3$ classes



- Here, $\pi_1 = \pi_2 = \pi_3 = 1/3$
- The dashed lines are known as the Bayes decision boundaries and the solid line is obtained using LDA

Alternative view of LDA (link with Fisher LDA)

- LDA can be used to perform *supervised dimensionality reduction*, by projecting the input data to a linear subspace consisting of the directions which maximize the separation between classes
  - We can interpret LDA as assigning x to the class whose mean is the closest in terms of Mahalanobis distance, while also accounting for the class prior probabilities
    - LDA is equivalent to first *sphering* (project) the data so that the covariance matrix is the identity, and then assigning x to the closest mean in terms of Euclidean distance
  - Note that the K means μ_k are vectors in R^p, and they lie in an affine subspace H of dimension at most $K-1$ (2 points lie on a line, 3 points lie on a plane, etc).
    - Computing Euclidean distances in original p-dimensional space is equivalent to first projecting the x into H, and computing the distances there
    - In other words, if x is closest to μ_k in the original space, it will also be the case in H. This shows that, implicit in the LDA classifier, there is a dimensionality reduction by linear projection onto a $K-1$ dimensional space

Alternative view of LDA (link with Fisher LDA)

Within class covariance matrix

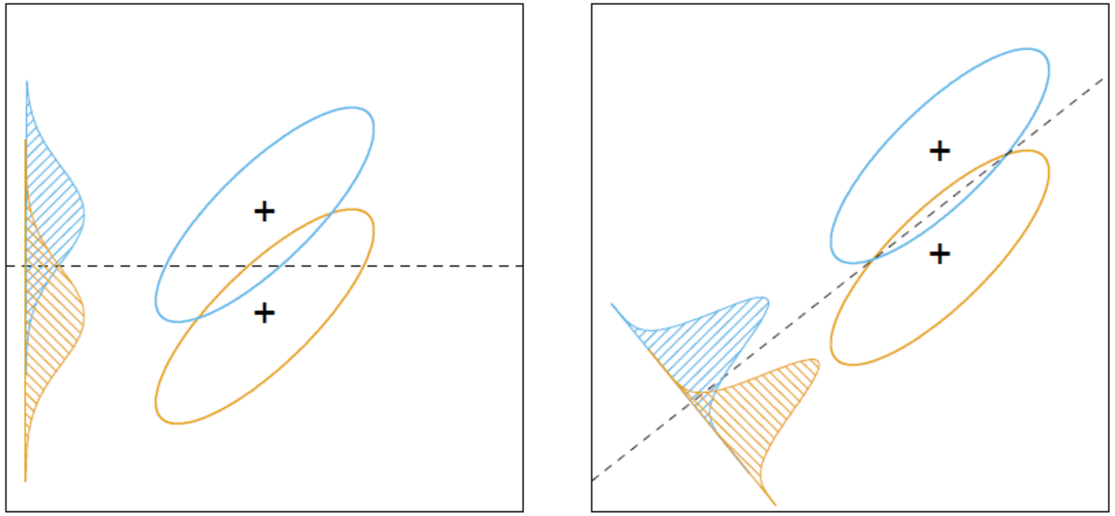
$$W = \sum_{k=1}^{K} \sum_{i \in C_k} (x_i - \mu_k)(x_i - \mu_k)^T$$

Between class covariance matrix

$$B = \sum_{k=1}^{K} n_k \, (\mu_k - \mu) \, (\mu_k - \mu)^T$$

- *Fisher criteria* is to maximize the *generalized Rayleigh quotient*

$$\max_a \frac{a^T B a}{a^T W a}$$

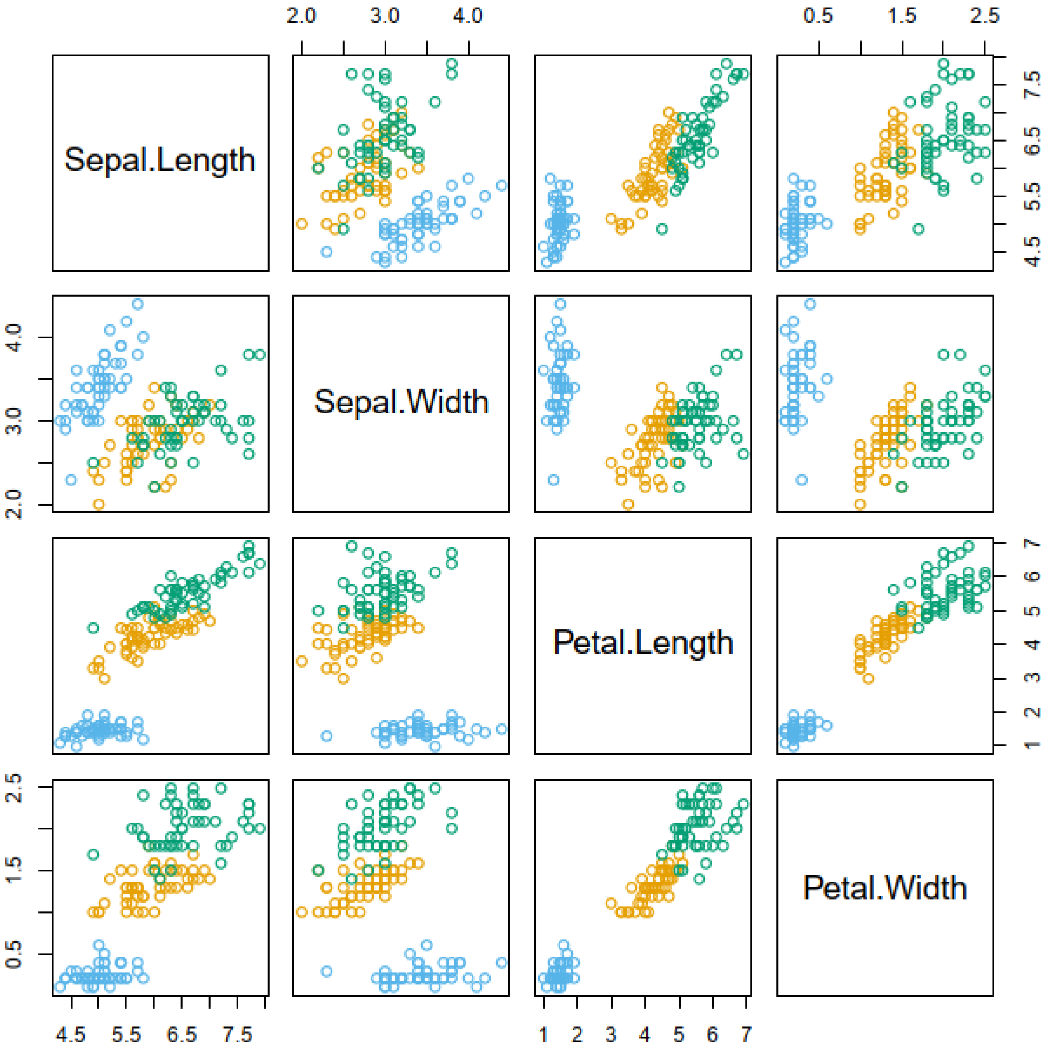


Alternative view of LDA (link with Fisher LDA)

- This is a generalize eigenvalue problem and a_1 is the eigenvector that correspond to the largest eigenvalue of $W^{-1}B$
  - One can find the next direction a_2 orthogonal to a_1 such that $\frac{{a_2}^T B a_2}{{a_2}^T W a_2}$ is maximize and it correspond to the second largest eigenvalue
  - a_l is known as *discriminant coordinate*, we can project the original data down to L dimension
  - Then we can classify the projected data $(\tilde{x}, \tilde{\mu}_k)$ using nearest to centroid rule $argmin_{j=1\ldots k} \frac{1}{2}|\tilde{x} - \tilde{\mu}_k|^2 - \log \tilde{\pi}_k$
    - This is equivalent to Maximum likelihood solution with Gaussian model subject to rank L (reduced rank LDA, ESL ch 4.3 and ESL exercise 4.8)

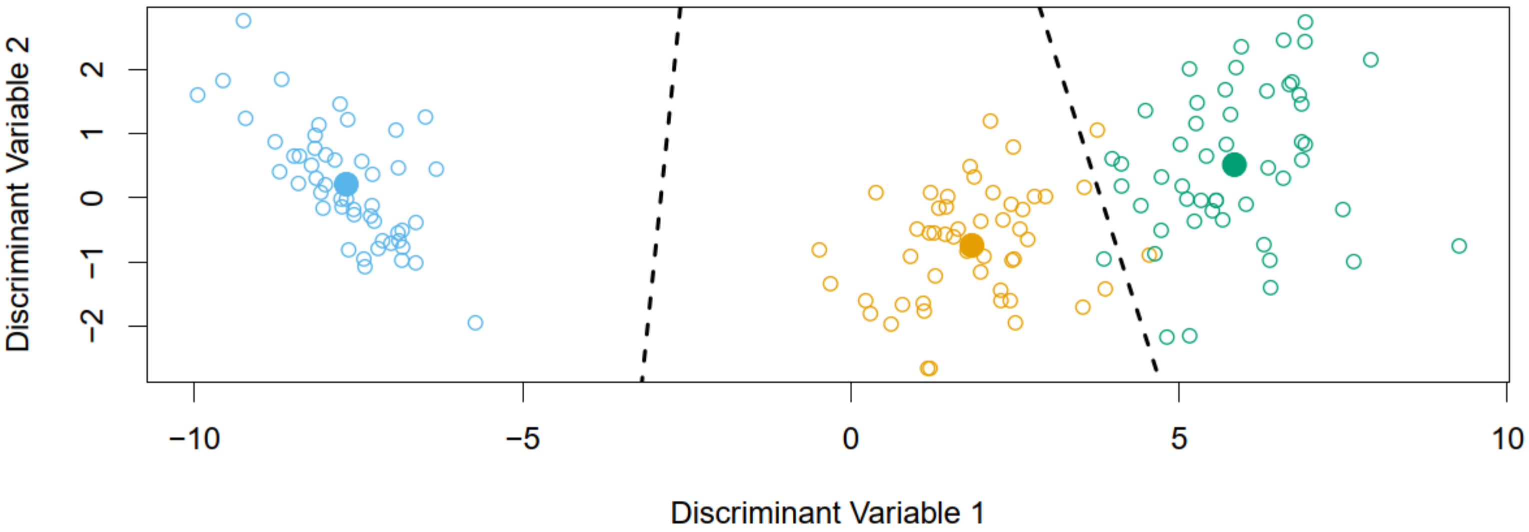
Fisher’s Iris Data

- 4 variables
- 3 species
- 50 samples/class
  - Setosa
  - Versicolor
  - Virginica
- LDA classifies all but 3 of the 150 training samples correctly



Fisher's Discriminant Plot

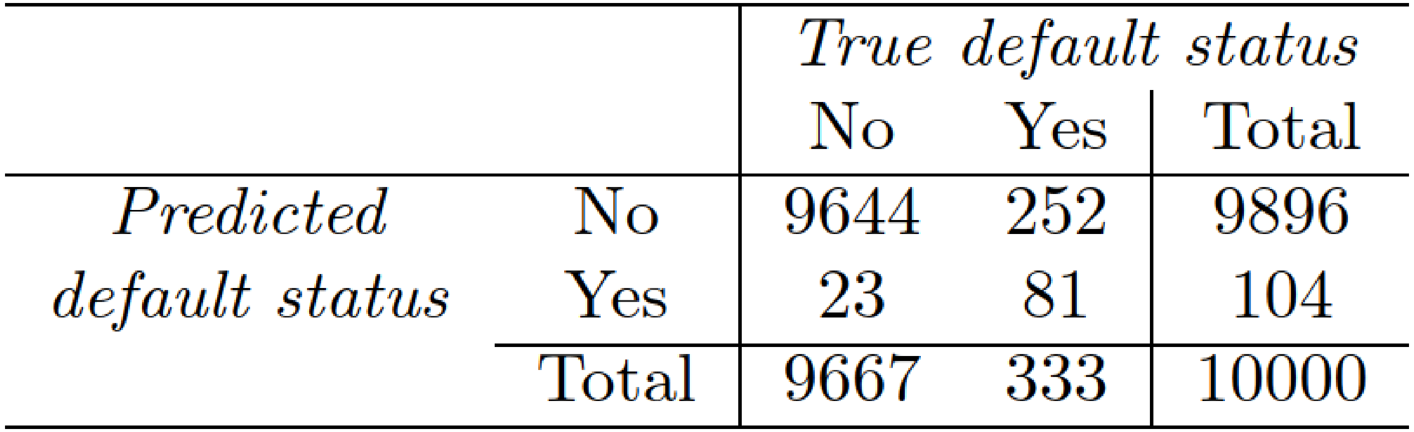
only two orange and one green are misclassified



- When there are K classes, linear discriminant analysis can be viewed exactly in a $K-1$ dimensional plot
  - Because it essentially classifies to the closest centroid, and they span a $K-1$ dimensional plane
  - Even when $K > 3$, we can find the "best" 2-dimensional plane for visualizing the discriminant rule

Back to the LDA on Credit Data

Use `balance` and `student` variables to build LDA



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

- $(23 + 252)/10{,}000$ errors — a 2.75% misclassification rate!
- Some caveats:
  1. This is training error, and we may be overfitting. Not a big concern here since $n = 10{,}000$ and $p = 2$!
  2. If we classified to the prior — always to class No in this case — we would make 333/10000 errors, or only 3.33%
  3. By the *confusion matrix*. Of the true No's, we make $23/9667 = 0.2\%$ errors; of the true Yes's, we make 252/333 = 75.7% errors

Types of errors

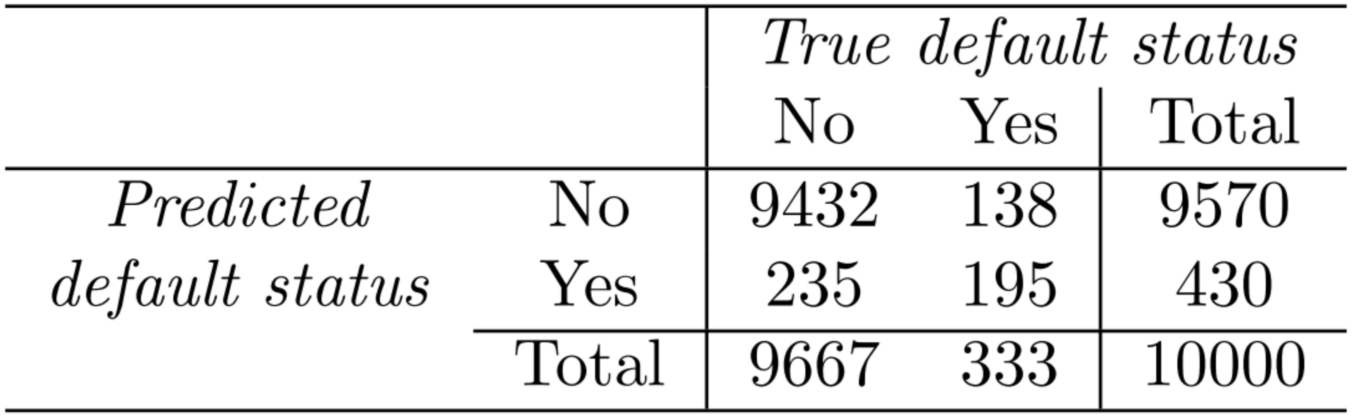
- False positive rate: The fraction of negative examples that are classified as positive — 0.2% in previous example
- False negative rate: The fraction of positive examples that are classified as negative — 75.7% in previous example
- We produced previous table (confusion matrix) by classifying to class Yes if

$$\widehat{Pr}(Default = Yes|Balance, Student) \ \geq \ 0.5$$

- We can change error rates by changing the threshold from 0.5 to some other value in [0, 1]:

$$\widehat{Pr}(Default = Yes|Balance, Student) \ \geq threshold$$

Threshold 0.2 here



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

FNR is 41.4%, although overall error rate increases to 3.73%

Varying the threshold

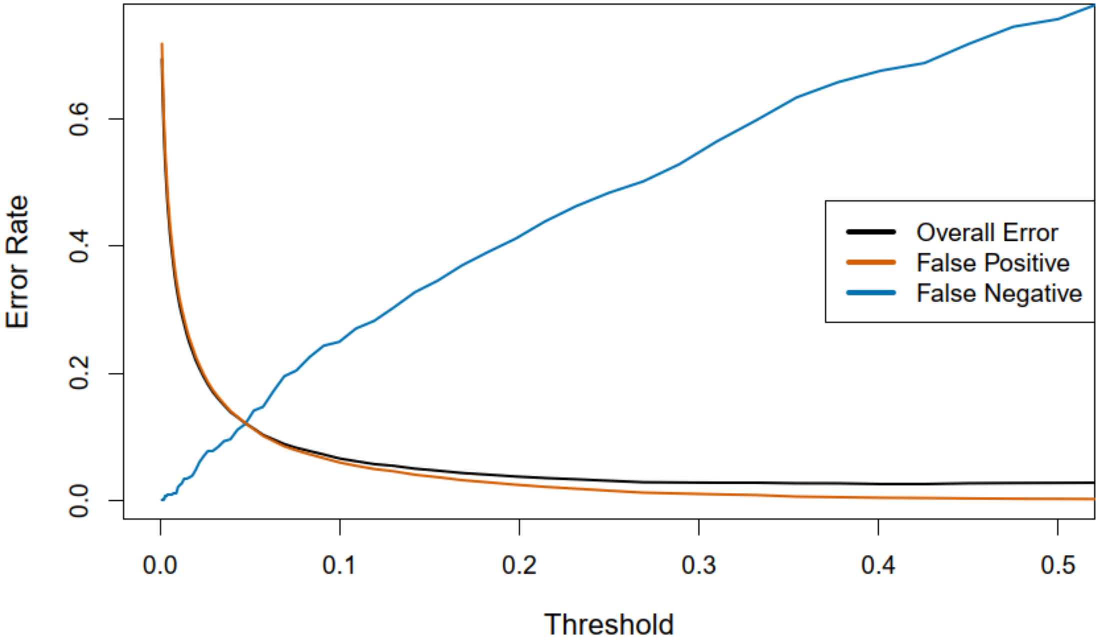
- In order to further reduce the false negative rate, we may want to reduce the threshold to 0.1 or less

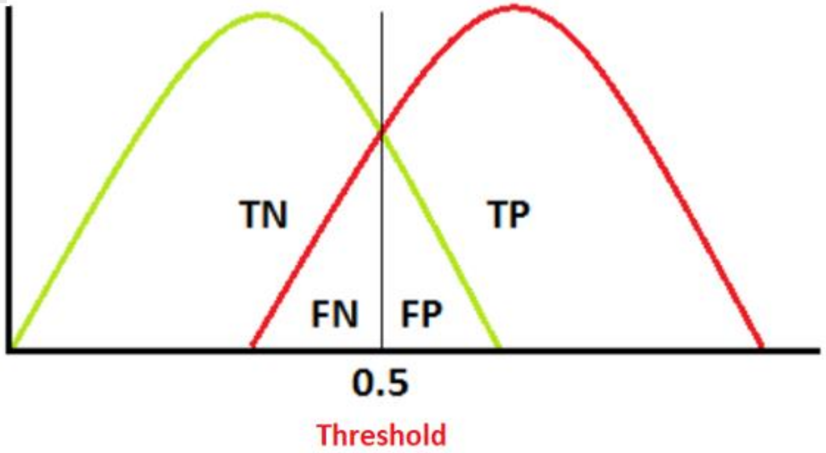
$$\text{False positive rate} = \frac{FP}{TN+FP}$$

$$\text{False negative rate} = \frac{FN}{TP+FN}$$

Confusion matrix

	Negative	Positive
Predict Negative	TN (True Negative)	FN (False Negative)
Predict Positive	FP (False Positive)	TP (True Positive)



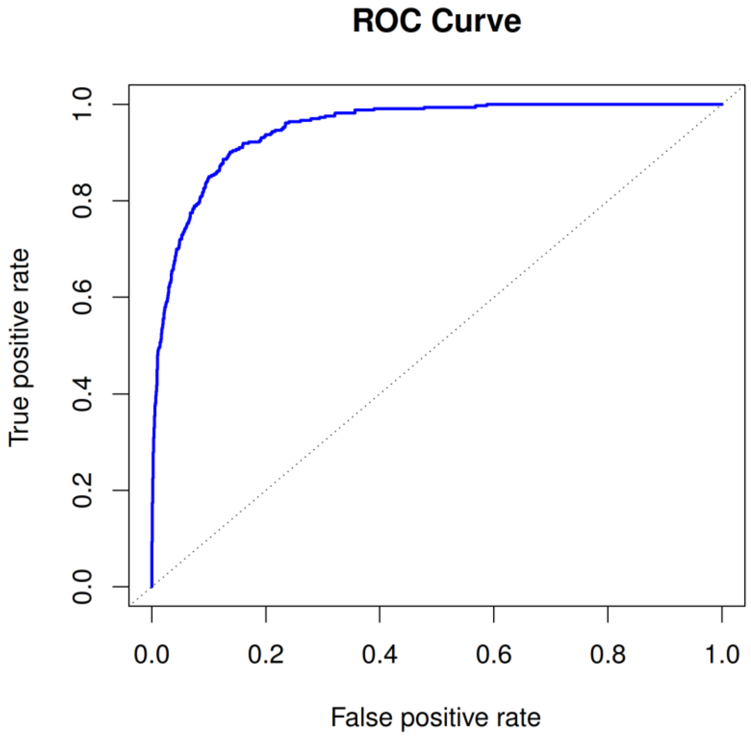


https://becominghuman.ai/whats-recall-and-precision-4a801b1ac0da

ROC (Receiver Operating Characteristics) Curve

	Negative	Positive
Predict Negative	TN	FN
Predict Positive	FP	TP

- The ROC plot displays both True and False positive rates simultaneously
  - $P = \frac{TP}{TP+FP}$ (Precision)
  - $R = \frac{TP}{TP+FN}$ (Recall) = Sensitivity = True positive rate = Power
  - Specificity = $\frac{TN}{TN+FP}$
  - False positive rate = $\frac{FP}{TN+FP}$ = 1- Specificity (Type I error)
  - False negative rate = $\frac{FN}{TP+FN}$ = 1- Sensitivity (Type II error)
  - Sometimes we use the AUC or area under the curve to summarize the overall performance. Higher AUC is good
  - Random classifier is the diagonal

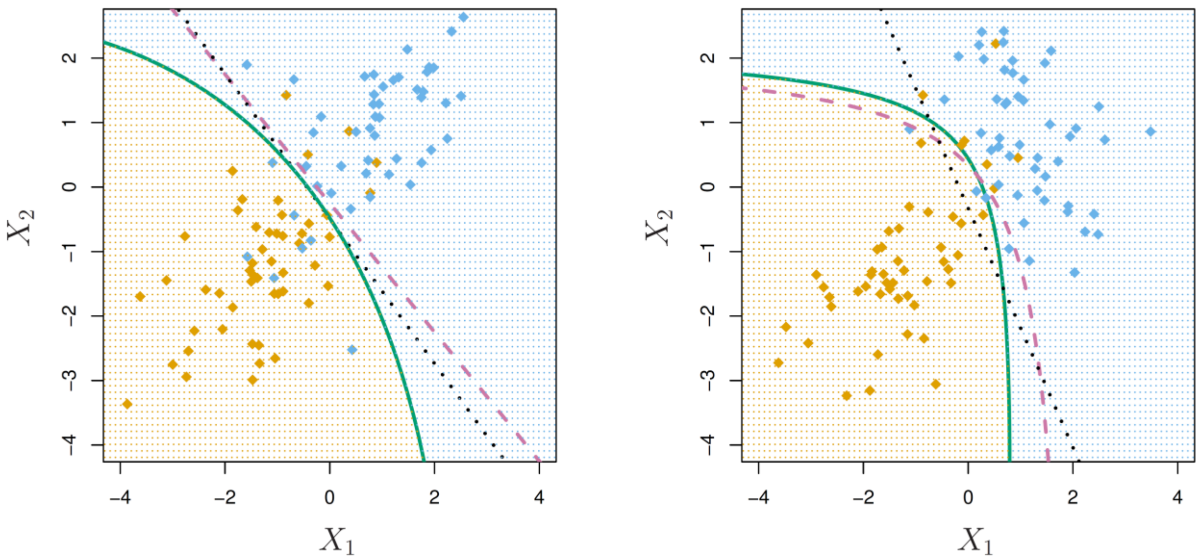


Other forms of Discriminant Analysis

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

- When $f_k(x)$ are Gaussian densities, with the same covariance matrix Σ in each class, this leads to linear discriminant analysis
  - By altering the forms for $f_k(x)$, we get different classifiers. With Gaussians *but different* Σ_k in each class, we get quadratic discriminant analysis (QDA)
    - It assumes that an observation from the kth class is of the form $X \sim N(\mu_k, \Sigma_k)$
  - With $f_k(x) = \prod_{j=1}^{p} f_{jk}(x)$ (conditional independence model) in each class we get naive Bayes. If Gaussian is also impose this will mean the Σ_k are diagonal
  - Many other forms, by proposing specific density models for $f_k(x)$, including nonparametric approaches

Quadratic Discriminant Analysis



The Bayes (purple dashed), LDA (black dotted), and QDA (green solid) decision boundaries for a two-class problem

- The Bayes classifier assigns an observation $X = x$ to which the following formula is largest

$$\delta_k(x) = -\frac{1}{2}x^T{\Sigma_k}^{-1}x + x^T{\Sigma_k}^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log(\pi_k)$$

the quantity x now appears as a quadratic function

Naive Bayes

- In general estimation of p-dimensional density $f_k(x)$ is challenging
- Assumes *features are independent in each class*

$$f_k(x) = f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)$$

  - It often leads to decent results, especially in settings where n is not large enough relative to p for us to effectively estimate the joint distribution of the predictors within each class
    - If X_j is quantitative, then we can assume that $X_j|Y = k \ \sim N(\mu_{kj}, \sigma_{kj}^2)$ which amounts to QDA with assumption that class-specific covariance matrix is diagonal. We can also replace $f_{kj}(x_j)$ with non-parametric estimate with probability mass function (histogram)
    - If X_j is qualitative, then we can simply count the proportion of training observations for the jth predictor corresponding to each class
    - The posterior probability is:

$$\Pr(Y = k|X = x) = \frac{\pi_k \times f_{k1}(x_1) \times f_{k2}(x_2) \times \cdots \times f_{kp}(x_p)}{\sum_{l=1}^{K} \pi_l \times f_{l1}(x_1) \times f_{l2}(x_2) \times \cdots \times f_{lp}(x_p)}$$

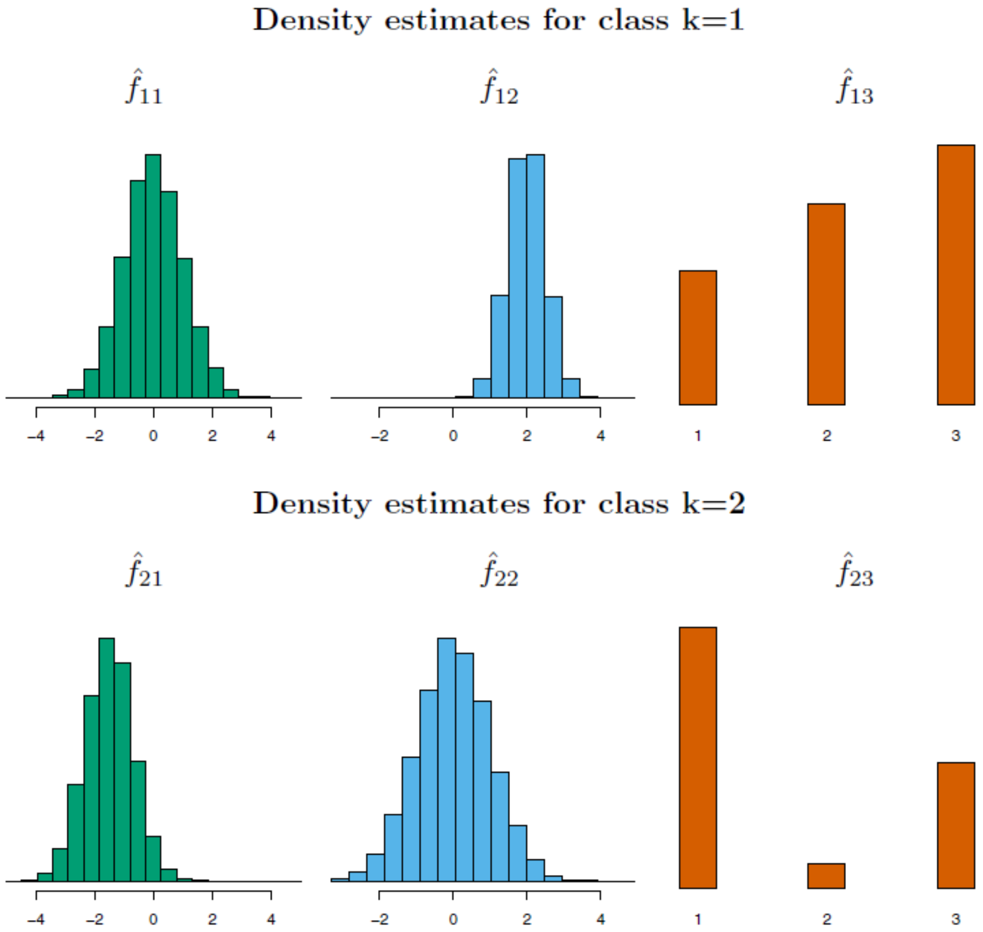
Naive Bayes

- We now consider the naive Bayes classifier in a toy example with $p = 3$ predictors and $K = 2$ classes. The first two predictors are quantitative, and the third predictor is qualitative with three levels. Suppose further that $\hat{\pi}_1 = \hat{\pi}_2 = 0.5$
  - New observation $x^* = (0.4,1.5,1)^T$,

$\hat{f}_{11}(0.4) = 0.368$, $\hat{f}_{12}(1.5) = 0.484$, $\hat{f}_{13}(1) = 0.226$, $\hat{f}_{21}(0.4) = 0.030$, $\hat{f}_{22}(1.5) = 0.130$, $\hat{f}_{23}(1) = 0.616$

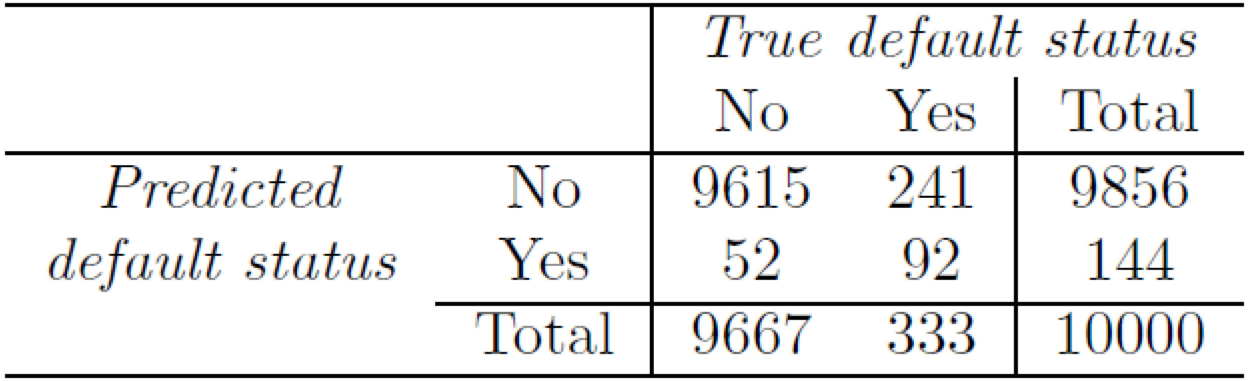
  - We have

$$Pr(Y = 1|X = x^*) = 0.944$$
$$Pr(Y = 2|X = x^*) = 0.056$$

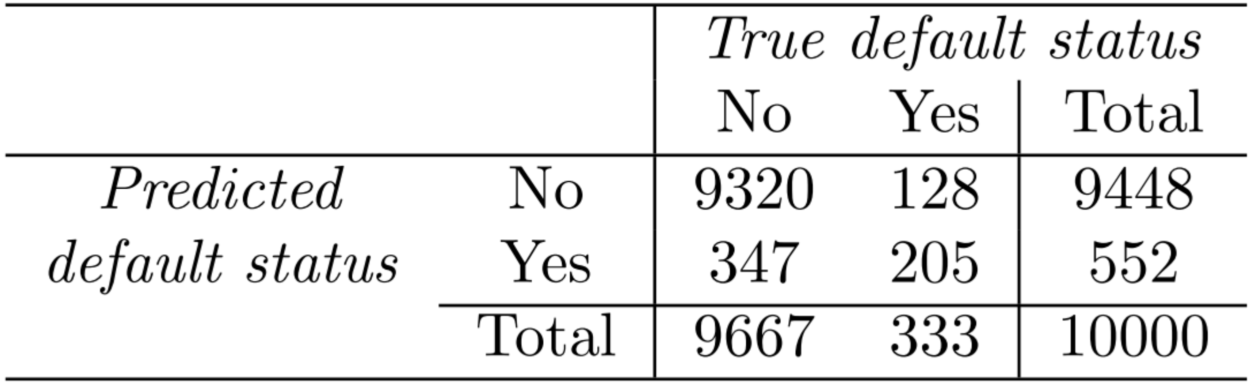


Naive Bayes

- Credit card data with threshold set to 0.5
  - We have assumed that each quantitative predictor is drawn from a Gaussian distribution



		True default status		
		No	Yes	Total
Predicted default status	No	9615	241	9856
	Yes	52	92	144
	Total	9667	333	10000



		True default status		
		No	Yes	Total
Predicted default status	No	9320	128	9448
	Yes	347	205	552
	Total	9667	333	10000

Threshold 0.2 here

  - Does not outperform LDA since $n = 10{,}000$ and $p = 2$ in this case

3. An Analytical Comparison of different methods

- We would like to assign an observation that maximizes the following formula

$$\log\left(\frac{\Pr(Y=k|X=x)}{\Pr(Y=K|X=x)}\right) \text{ for } \quad k = 1, \dots, K$$

- For LDA we have

$$\log\left(\frac{\Pr(Y=k|X=x)}{\Pr(Y=K|X=x)}\right) = \log(\frac{\pi_k}{\pi_K}) - \frac{1}{2}(\mu_k + \mu_K)^T\Sigma^{-1}(\mu_k - \mu_K) + x^T\Sigma^{-1}(\mu_k - \mu_K)$$

$$= a_k + \textstyle\sum_{j=1}^{p} b_{kj}x_j$$

  - So LDA, like logistic regression, assumes that log odds of the posterior probabilities is linear in x

- For QDA we have

$$\log\left(\frac{\Pr(Y=k|X=x)}{\Pr(Y=K|X=x)}\right) = a_k + \sum_{j=1}^{p} b_{kj}x_j + \sum_{j=1}^{p}\sum_{l=1}^{p} c_{kjl}\, x_j x_l$$

  - QDA assumes that the log odds of the posterior probabilities is quadratic in x

An Analytical Comparison of different methods

- For naïve Bayes

$$\log\left(\frac{\Pr(Y=k|X=x)}{\Pr(Y=K|X=x)}\right) = \log\left(\frac{\pi_k}{\pi_K}\right) + \sum_{j=1}^{p} \log(\frac{f_{kj}(x_j)}{f_{Kj}(x_j)}) = a_k + \sum_{j=1}^{p} g_{kj}(x_j)$$

1. LDA is a special case of QDA with $c_{kjl} = 0$ for all $j = 1, \ldots, p, l = 1, \ldots, p,$ and $k = 1, \ldots, K$
2. Any classifier with a linear decision boundary can be link to naïve Bayes with $g_{kj}(x_j) = b_{kj}x_j$ and can be considered as a special case of naïve Bayes
    - If we model $f_{kj}(x_j)$ in the naive Bayes classifier using a one-dimensional Gaussian distribution $N(\mu_{kj}, \sigma_j^2)$, then we end up with $g_{kj}(x_j) = b_{kj}x_j$, where $b_{kj} = (\mu_{kj} - \mu_{Kj})/\sigma_j^2$ In this case, naive Bayes is actually a special case of LDA with Σ restricted to be a diagonal matrix with jth diagonal element equal to σ_j^2
3. QDA and naive Bayes can produce flexible fit

An Analytical Comparison of different methods

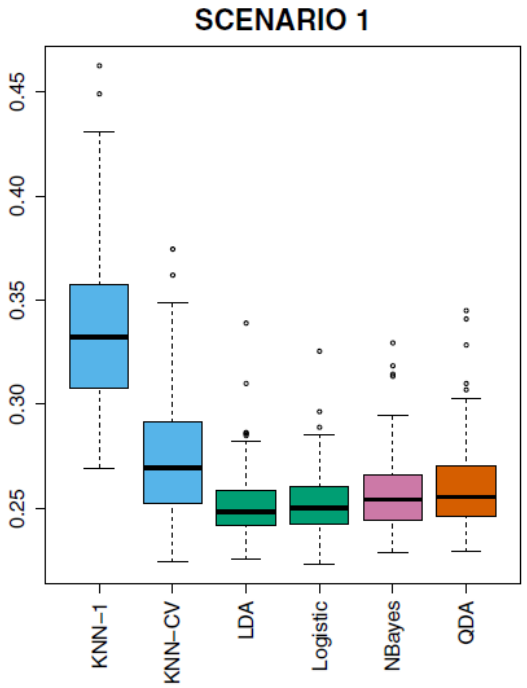
- LDA would do better than Logistic Regression if the assumption of normality hold, otherwise logistic regression can outperform LDA
- KNN is completely non-parametric: No assumptions are made about the shape of the decision boundary!
  - In order to provide accurate classification, KNN requires a lot of observations relative to the number of predictors. This has to do with the fact that KNN is non-parametric, and thus tends to reduce the bias while incurring a lot of variance
- QDA is a compromise between non-parametric KNN method and the linear LDA and logistic regression. If the true decision boundary is:
  - Linear: LDA and Logistic outperforms
  - Moderately Non-linear: QDA outperforms
  - More complicated: KNN is superior

An Empirical Comparison

- We generated data from six different scenarios. each of which involves a binary (two-class) classification problem. There are two quantitative predictors
  1. In three of the scenarios, the Bayes decision boundary is linear, and in the remaining scenarios it is non-linear
  2. For each scenario, we produced 100 random training data sets. On each of these training sets, we fit each method to the data and computed the resulting *test error rate* on a large test set
  3. The KNN method requires selection of K, the number of neighbors. We performed KNN with two values of K: $K = 1$, and a value of K that was chosen automatically using an approach called cross-validation, which we discuss further in Chapter 5
  4. We applied naive Bayes assuming univariate Gaussian densities for the features within each class

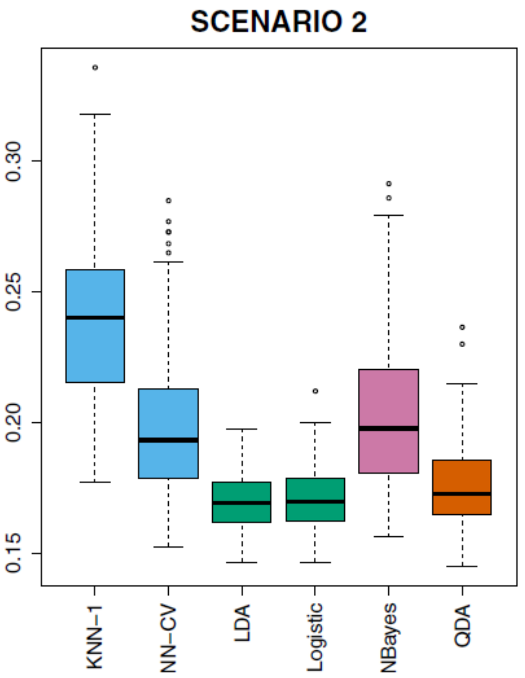
An Empirical Comparison

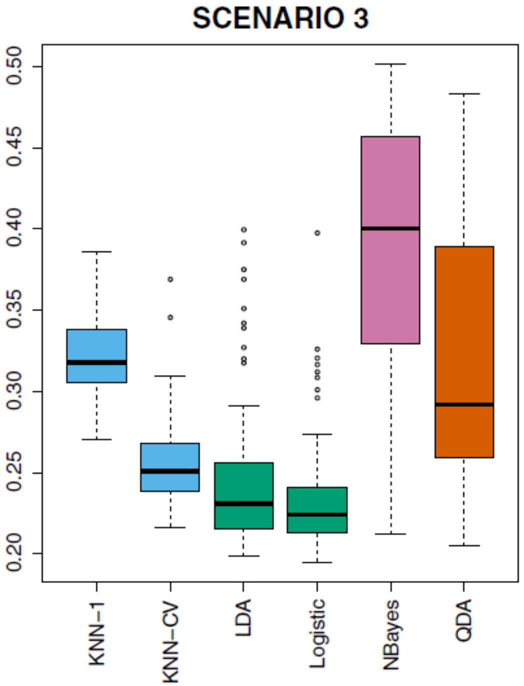
1. There were 20 training observations in each of the two classes. The observations within each class were *uncorrelated random normal* variables with a *different mean* in each class
   - The left-hand panel shows that LDA performed well in this setting, as one would expect since this is the model assumed by LDA. Logistic regression assumes a linear decision boundary, its results were only slightly inferior to those of LDA
   - KNN performed poorly because it paid a price in terms of variance that was not offset by a reduction in bias. QDA also performed worse than LDA, since it fits a more flexible classier than necessary
   - Naive Bayes was slightly better than QDA because the naive Bayes assumption of independent predictors is correct



An Empirical Comparison

2. Details are as in Scenario 1, except that within each class, the two predictors had *a correlation of* -0.5
   - The notable exception is naive Bayes, which performs very poorly here, since the naive Bayes assumption of independent predictors is violated
3. As in 2 but we here generated X_1 and X_2 from the *multivariate t-distribution*, with 50 observations per class
   - The decision boundary was still linear, and so fit into the logistic regression
   - The set-up violated the assumptions of LDA. It shows that logistic regression outperformed LDA, though both methods were superior to the other approaches
   - In particular, the QDA results deteriorated considerably as a consequence of non-normality
   - Naive Bayes performed very poorly because the independence assumption is violated

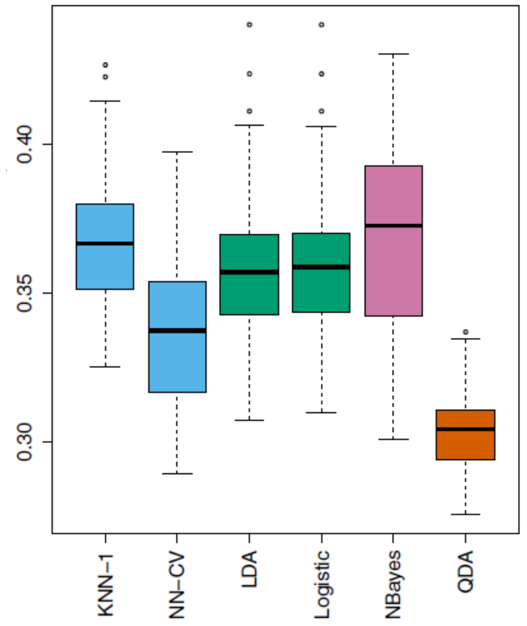

SCENARIO 2


SCENARIO 3

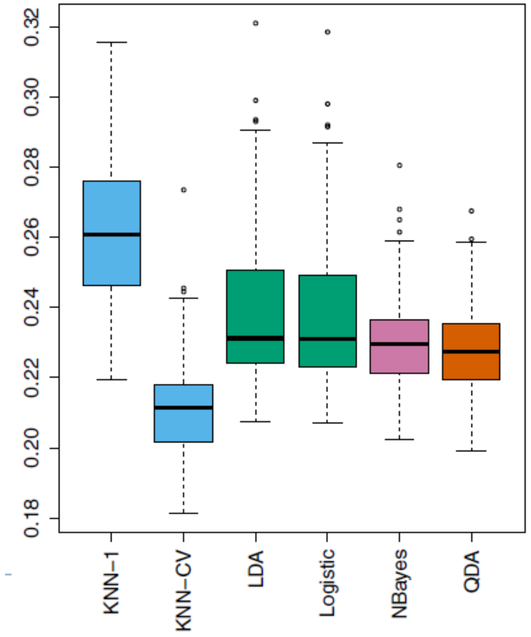
An Empirical Comparison

4. The data were generated from a *normal distribution*, with a correlation of 0.5 between the predictors in first class, and correlation of -0.5 between the predictors in the second class
   - This setup corresponded to the QDA assumption and resulted in quadratic decision boundaries
   - The naive Bayes assumption of independent predictors is violated therefore performs poorly
5. Within each class, the observations were generated from a normal distribution with uncorrelated predictors. However, the responses were sampled from the logistic function applied to a *non-linear function* of the predictors
   - QDA and naive Bayes gave slightly better results than the linear methods while the much more flexible KNN-CV method gave the best results
   - But KNN with $K = 1$ gave the worst results out of all methods

SCENARIO 4

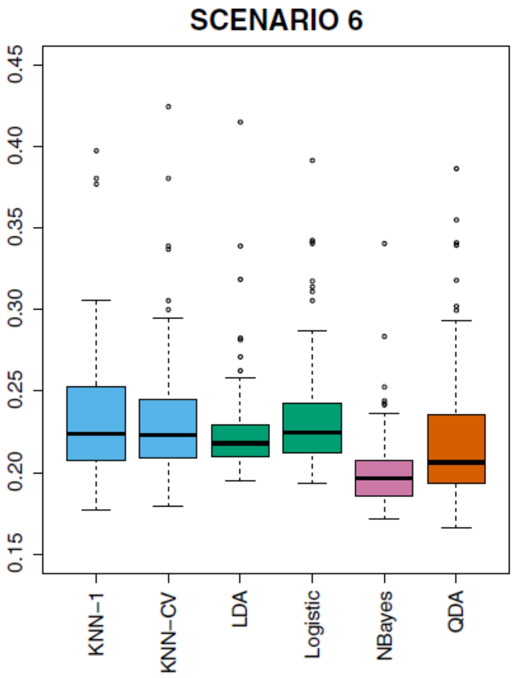


SCENARIO 5



An Empirical Comparison

6. The observations were generated from a normal distribution with a different diagonal covariance matrix for each class. However, *the sample size was very small*: just $n = 6$ in each class
   - Naive Bayes performed very well, because its assumptions are met. LDA and logistic regression performed poorly because the true decision boundary is non-linear, due to the unequal covariance matrices
   - QDA performed a bit worse than naïve Bayes, because given the very small sample size, the former incurred too much variance in estimating the correlation between the predictors within each class. KNN’s performance also suffered due to the very small sample size

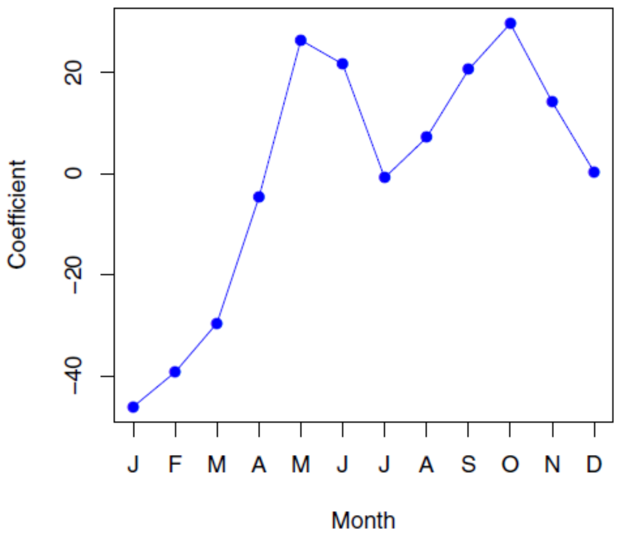


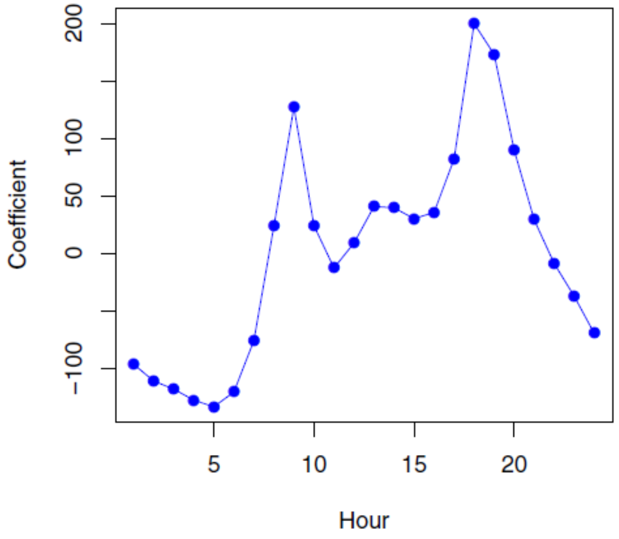
Bikeshare dataset

- We consider the Bikeshare data set. The response is `bikers`, the number of hourly users of a bike sharing program in Washington, DC
  - This response value is hard to be classified into qualitative or quantitative variable: it takes on non-negative integer values, or counts
  - We will consider *counts predicting bikers* using the covariates `mnth` (month of the year), `hr` (hour of the day, from 0 to 23), `workingday` (an indicator variable that equals 1 if it is neither a weekend nor a holiday), `temp` (the normalized temperature, in Celsius), and `weathersit` (a qualitative variable that takes on one of four possible values: clear; misty or cloudy; light rain or light snow; or heavy rain or heavy snow)

Linear Regression on the Bikeshare Data

	Coefficient	Std. error	t-statistic	p-value
`Intercept`	73.60	5.13	14.34	0.00
`workingday`	1.27	1.78	0.71	0.48
`temp`	157.21	10.26	15.32	0.00
`weathersit[cloudy/misty]`	-12.89	1.96	-6.56	0.00
`weathersit[light rain/snow]`	-66.49	2.97	-22.43	0.00
`weathersit[heavy rain/snow]`	-109.75	76.67	-1.43	0.15

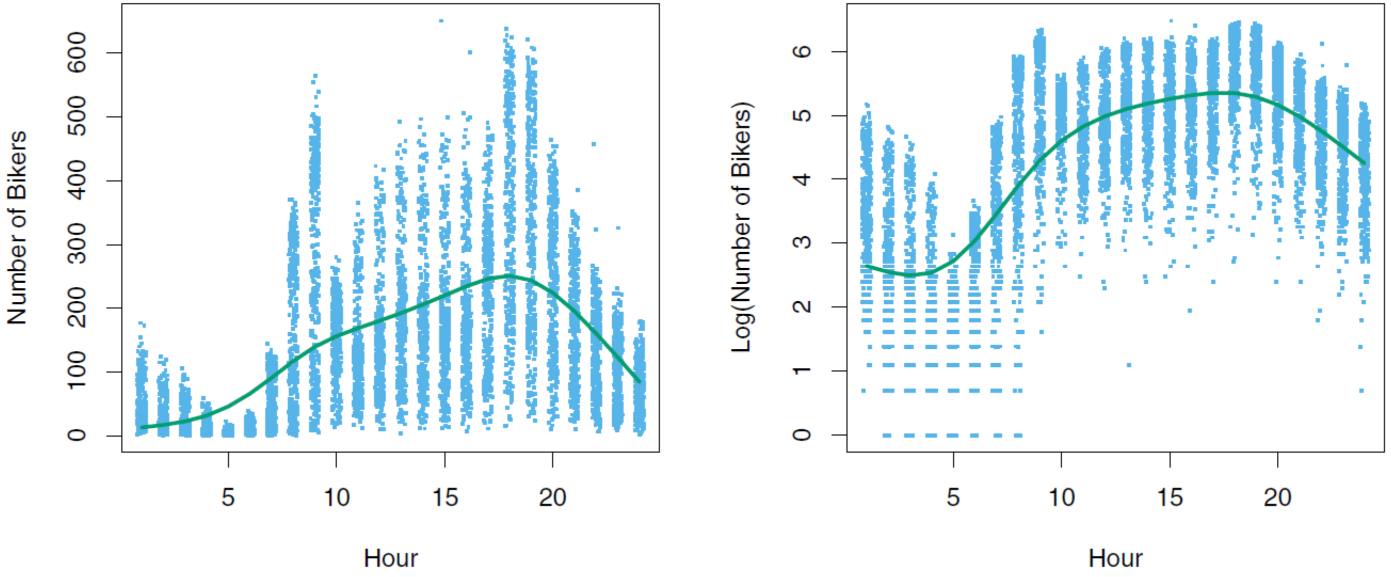




Bikeshare dataset

- Upon more careful inspection, some issues become apparent.

1. For example, 9.6% of the fitted values in the Bikeshare data set are negative: that is, the linear regression model predicts a negative number of users during 9.6% of the hours in the data set
2. The variance is not constant as well
3. The response Y is necessarily continuous valued (quantitative). Thus, the integer nature of the response bikers suggests that a linear regression model is not entirely satisfactory for this data set



4. Poisson Regression

- Suppose that a random variable Y takes on nonnegative integer values, i.e. $Y \in \{0, 1, 2, \dots\}$. If Y follows the Poisson distribution, then

$$\Pr(Y = k) = \frac{e^{-\lambda}\lambda^k}{k!} \text{ for } k = 0,1,2, \dots$$

  - Here, $\lambda > 0$ and $\lambda = E(Y) = Var(Y)$, The Poisson distribution is typically used to model counts; this is a natural choice for a number of reasons

- We consider the following model for the mean $\lambda = E(Y|X)$

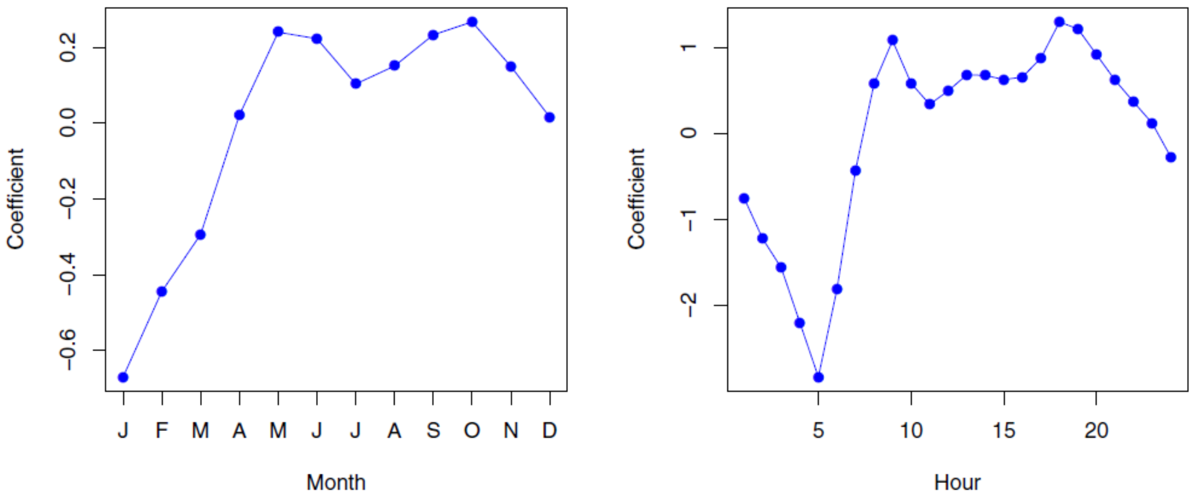
$$Y|X = \ Poisson(\lambda)$$
$$\log(\lambda(X_1, \dots, X_p)) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

- Given n independent observations from the Poisson regression model, the likelihood takes the form

$$l(\beta_0, \beta_1, \dots, \beta_p) = \prod_{i=1}^{n} \frac{e^{-\lambda(x_i)}\lambda(x_i)^{y_i}}{y_i!}$$

Poisson Regression on the Bikeshare Data

	Coefficient	Std. error	z-statistic	p-value
`Intercept`	4.12	0.01	683.96	0.00
`workingday`	0.01	0.00	7.5	0.00
`temp`	0.79	0.01	68.43	0.00
`weathersit[cloudy/misty]`	-0.08	0.00	-34.53	0.00
`weathersit[light rain/snow]`	-0.58	0.00	-141.91	0.00
`weathersit[heavy rain/snow]`	-0.93	0.17	-5.55	0.00



Generalized Linear Models (GLM) in Greater Generality

- We have now discussed three types of regression models: linear, logistic and Poisson. These approaches share some common characteristics:

1. Each approach uses predictors $X_1, \ldots, X_P$ to predict a response Y. We assume that, conditional on $X_1, \ldots, X_P$, Y belongs to a certain family of distributions. For linear regression, we typically assume that $Y|X$ follows a Gaussian or normal distribution. For logistic regression, we assume that $Y|X$ follows a Bernoulli distribution. Finally, for Poisson regression, we assume that $Y|X$ follows a Poisson distribution
2. Each approach models the *mean* of Y as a function of the predictors.

$$E\left(Y\middle|X_1, \ldots, X_p\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

$$E\left(Y\middle|X_1, \ldots, X_p\right) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}$$

$$E\left(Y\middle|X_1, \ldots, X_p\right) = e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}$$

Generalized Linear Models in Greater Generality

- They can be expressed using a *link function*, η, which link function applies a transformation to $E\left(Y\middle|X_1, \ldots, X_p\right)$ so that the transformed mean is a linear function of the predictors. That is

$$\eta\left(E\left(Y\middle|X_1, \ldots, X_p\right)\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

  - The link functions for linear, logistic and Poisson regression are $\eta(\mu) = \mu$, $\eta(\mu) = \log(\mu/(1-\mu))$, and $\eta(\mu) \;=\; \log(\mu)$, respectively
- The Gaussian, Bernoulli and Poisson distributions are all members of a wider class of distributions, known as the exponential family
  - In general, we can perform a regression by modeling the response Y as coming from a particular member of the exponential family, and then transforming the mean of the response so that the transformed mean is a linear function of the predictors via the link function

Appendix

GLM vs linear regression

- GLM model the mean
  - Variance is related to mean and error not i.i.d.
    - https://stats.stackexchange.com/questions/401045/why-no-variance-term-in-bayesian-logistic-regression
    - https://stats.stackexchange.com/questions/259704/is-there-i-i-d-assumption-on-logistic-regression
- Think of it as modeling the conditional distribution
    - https://stats.stackexchange.com/questions/55538/does-poisson-regression-have-an-error-term
    - https://stats.stackexchange.com/questions/124818/logistic-regression-error-term-and-its-distribution
    - https://stats.stackexchange.com/questions/353231/conditional-distribution-in-logistic-regression

Coefficient and Standard error

- Coefficient and Standard error
  - Logistic regression
    - https://web.stanford.edu/class/archive/stats/stats200/stats200.1172/Lecture26.pdf
    - https://stats.stackexchange.com/questions/303180/standard-error-of-the-estimate-in-logistic-regression
    - https://stats.stackexchange.com/questions/68080/basic-question-about-fisher-information-matrix-and-relationship-to-hessian-and-s
  - LDA and QDA
    - https://arxiv.org/pdf/1906.02590.pdf
  - GLM
    - https://www.sagepub.com/sites/default/files/upm-binaries/21121_Chapter_15.pdf
  - OVO or OVR
    - https://en.wikipedia.org/wiki/Multiclass_classification

Review of Covariance Matrix

- Let $x_1, \ldots, x_n$ be length-p observation vectors

$$x_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix}$$

- Without Loss Of Generality (WLOG), let their mean be length-p 0-vector
- Let the data matrix $X = (x_1, x_2, \ldots, x_n)$ be a p by n matrix
- The sample covariance matrix

$$S = XX^T/(n-1) = \sum_{i=1}^{n} x_i x_i^T/(n-1) = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T/(n-1)$$

Review of eigenvalue decomposition- Maximum variance formulation

- Find a direction vector $u_1 \in R^p$ and $u_1^T u_1 = 1$ such that the variance of the projected data is maximized

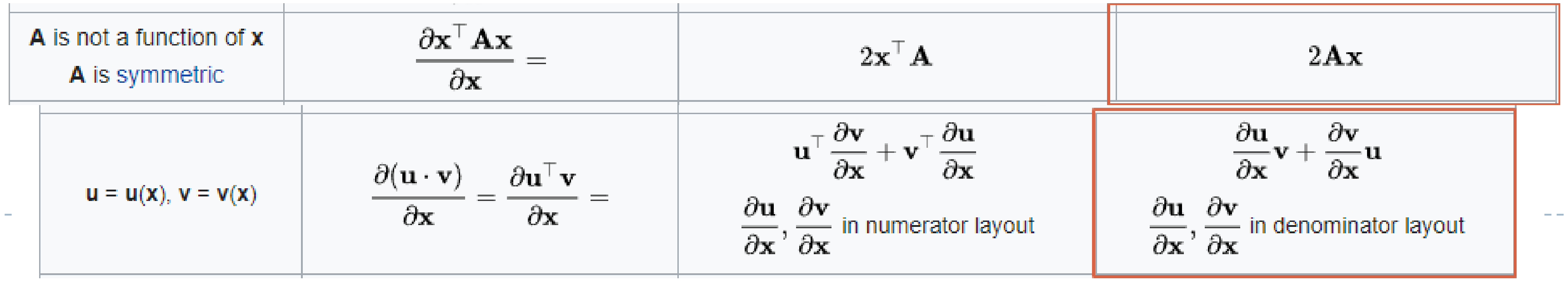
$$\frac{1}{n}\sum_{i=1}^{n}(u_1^T x_i - u_1^T \bar{x})^2 = u_1^T S u_1$$

  - To enforce the constraint, we introduce a Lagrange multiplier denoted by λ_1 and get the unconstrained maximization of

$$u_1^T S u_1 + \lambda_1(1 - u_1^T u_1) \text{ or maximize } \frac{u^T S u}{u^T u}$$

  - By setting the derivative with respect to u_1 equal to zero, we see that this quantity will have a stationary point when

$$S u_1 = \lambda_1 u_1$$



A is not a function of **x** **A** is symmetric	$\frac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$	$2\mathbf{x}^\top\mathbf{A}$	$\boxed{2\mathbf{A}\mathbf{x}}$
u = **u**(**x**), **v** = **v**(**x**)	$\frac{\partial(\mathbf{u}\cdot\mathbf{v})}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}^\top\mathbf{v}}{\partial \mathbf{x}} =$	$\mathbf{u}^\top\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{v}^\top\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$ in numerator layout	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\mathbf{v} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}\mathbf{u}$ $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}, \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$ in denominator layout

Review of eigenvalue decomposition- Maximum variance formulation

- u_1 must be an eigenvector of S, if we left-multiply by u_1^T we get

$$u_1^T S u_1 = \lambda_1$$

  - and so the variance will be a maximum when we set u_1 equal to the eigenvector having the largest eigenvalue λ_1. This eigenvector is known as the first principal component.

- We can define additional principal components in an incremental fashion by choosing each new direction to be that which maximizes the projected variance amongst all possible directions orthogonal to those already considered.
  - In a r-dimensional projection space, we now consider the optimal linear projection for which the variance of the projected data is maximized is defined by the r eigenvectors $u_1, \ldots, u_r$ of the data covariance matrix S corresponding to the r largest eigenvalues $\lambda_1, \ldots, \lambda_r$.

Principal Component Analysis (PCA) (1/2)

- If we collect eigenvectors and eigenvalues into matrix

$$S_{p\times p}U_{p\times p} \;= U_{p\times p}\Lambda_{p\times p}$$
$$S_{p\times p} = U_{p\times p}\Lambda_{p\times p}U_{p\times p}^{T}$$

- Note $X = USV^T$
  - Scores are $U^TX = SV^T$
- It is equivalent to Minimum error formulation

$$argmin_{U\epsilon\, O_{p,r}} \sum_{i=1}^{n} |(X_i - \bar{X}) - UU^T(X_i - \bar{X}))|_F^2$$

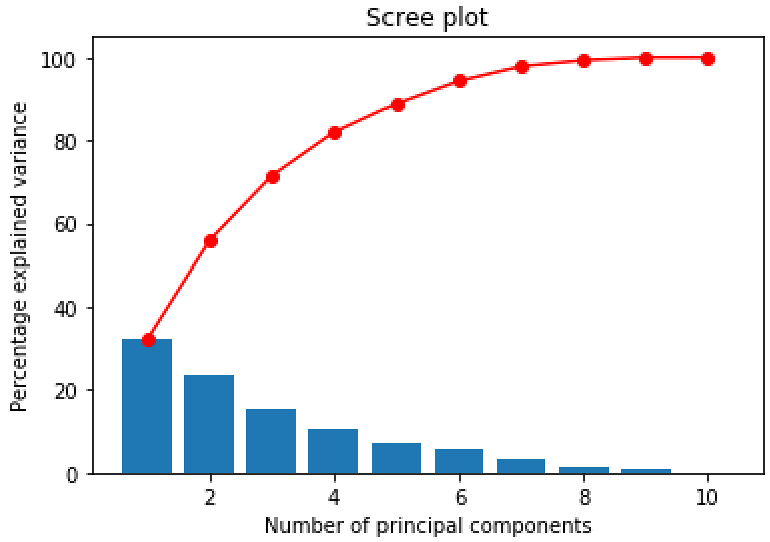
	Convention 1	Convention 2
U	Principal component Principal direction Loading	Principal axis Principal direction
U^TX	Principal component scores	Principal component

Principal Component Analysis (PCA) (2/2)

- Connection with SVD

$$\mathrm{S} = \frac{XX^T}{n-1} = \frac{UDV^TVDU^T}{n-1} = U\frac{D^2}{n-1}U^T = U\Lambda U^T$$

- In practice, we will often scale data before PCA



- Whiten data matrix (identity covariance matrix)
  - $\Lambda^{-1/2}U^T\mathrm{X}$
- ZCA (Close to original data (often not reduce dimension))
  - $U\Lambda^{-1/2}U^T\mathrm{X}$