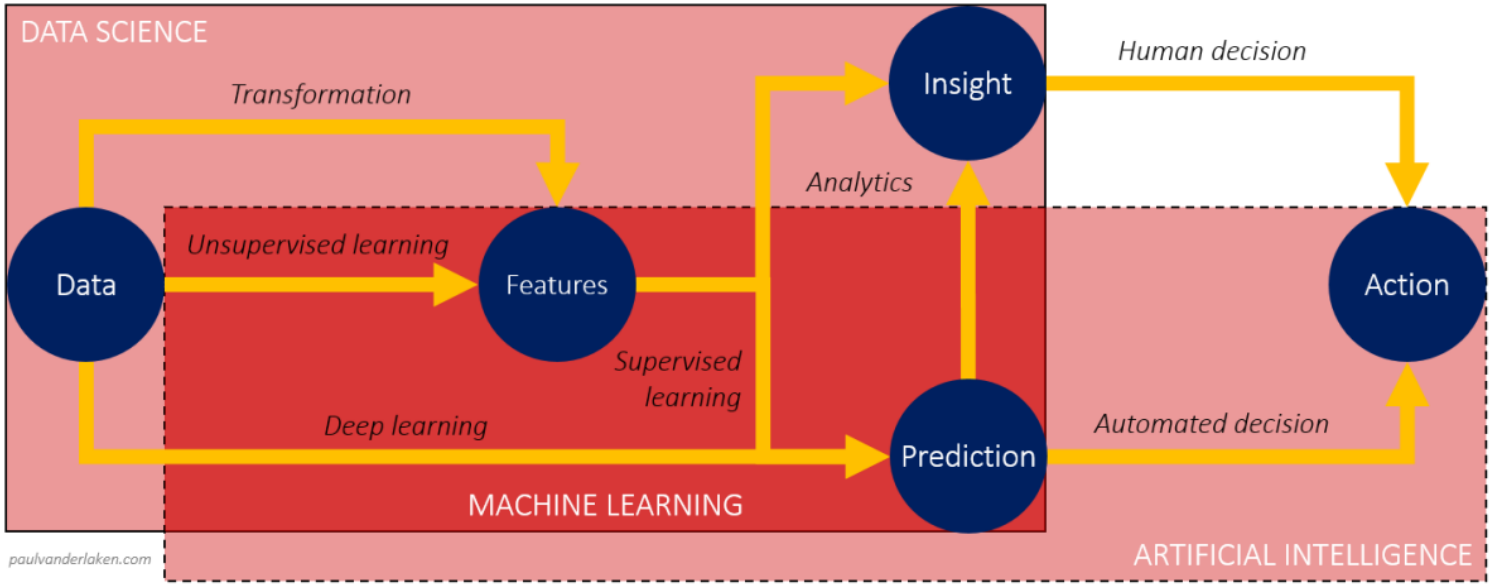
Supervised Learning for Classification



Classification

- Predicting a *qualitative* response for an observation can be referred to as *classifying* that observation, since it involves assigning the observation to a category, or class.

- Thus, *classification models* are supervised learning methods for which the true class labels for the data points are given in the training data.

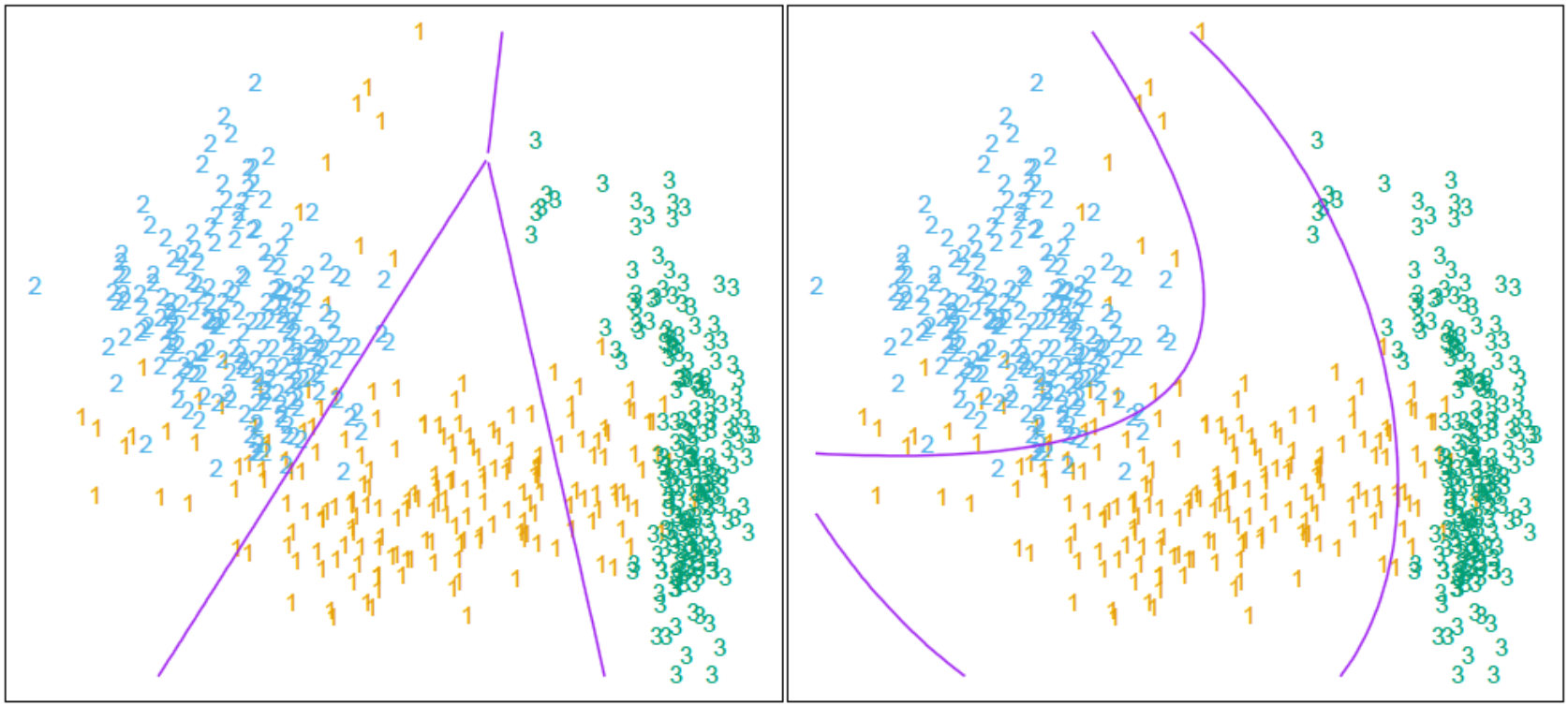
- The methods used for classification often predict the probability of each of the categories of a qualitative variable as the basis for making the classification decision.

Classification Setting

- Training data: $\{(x_1,g_1), (x_2,g_2),\ldots,(x_N,g_N)\}$
- The feature vector $X = (X_1, X_2,\ldots,X_p)$, where each X_j is quantitative.
- The response variable G is categorical s.t. $G \in G = \{1,2, \ldots, K\}$
- Form a predictor $G(x)$ to predict G based on X.
- Note that $G(x)$ divides the input space (feature vector space) into a collection of regions, each labeled by one class.

Classification Setting (cont.)

- For each plot, the feature vector space is divided into three pieces, each assigned with a particular class.



Classification Error Rate

- The *classification error rate* is the number of observations that are misclassified over the sample size:

$$\frac{1}{n}\sum_{i=1}^{n} I(\hat{Y}_i \neq Y_i)$$

where $I(\hat{Y}_i \neq Y_i) = 1$ if $\hat{Y}_i \neq Y_i$ and 0 otherwise.

- For binary classification let $\hat{Y}$ be a 0-1 vector of the predicted class labels and *Y* be a 0-1 vector of the observed class labels.

Naïve Bayes Classification

Probability Basics

- Prior, conditional and joint probability for random variables
  - Prior probability: $P(X)$
  - Conditional probability: $P(X_1 \mid X_2), P(X_2 \mid X_1)$
  - Joint probability: $\mathbf{X} = (X_1, X_2), P(\mathbf{X}) = P(X_1, X_2)$
  - Relationship: $P(X_1, X_2) = P(X_2 \mid X_1)P(X_1) = P(X_1 \mid X_2)P(X_2)$
  - Independence: $P(X_2 \mid X_1) = P(X_2), P(X_1 \mid X_2) = P(X_1), P(X_1, X_2) = P(X_1)P(X_2)$

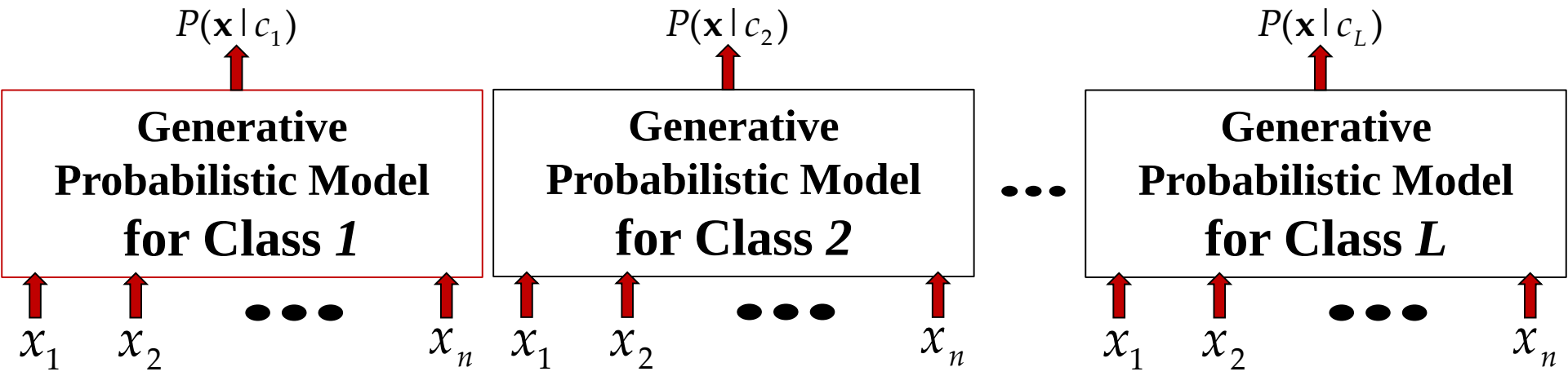
- **Bayes' Rule**

$$P(C \mid \mathbf{X}) = \frac{P(\mathbf{X} \mid C)P(C)}{P(\mathbf{X})} \quad Posterior = \frac{Likelihood \times Prior}{Evidence}$$

Probabilistic Classification

- Establishing a probabilistic model for classification
  - **Generative model**

$$P(\mathbf{X} \mid C) \quad C = c_1, \cdots, c_L, \mathbf{X} = (X_1, \cdots, X_n)$$



$$\mathbf{x} = (x_1, x_2, \cdots, x_n)$$

Probabilistic Classification (cont.)

- MAP classification rule
  - **MAP**: **M**aximum **A** **P**osterior
  - Assign ***x*** to *c** if

$$P(C = c^* \mid \mathbf{X} = \mathbf{x}) > P(C = c \mid \mathbf{X} = \mathbf{x}) \quad c \neq c^*,\ c = c_1, \cdots, c_L$$

- Generative classification with the MAP rule
  - Apply Bayes' rule to convert them into posterior probabilities

$$P(C = c_i \mid \mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X} = \mathbf{x} \mid C = c_i) P(C = c_i)}{P(\mathbf{X} = \mathbf{x})}$$
$$\propto P(\mathbf{X} = \mathbf{x} \mid C = c_i) P(C = c_i)$$
$$\text{for } \ i = 1, 2, \cdots, L$$

  - Then apply the MAP rule

Naïve Bayes Classifier

- Bayes classification

$$P(C \mid \mathbf{X}) \propto P(\mathbf{X} \mid C)P(C) = P(X_1, \cdots, X_n \mid C)P(C)$$

Difficulty: learning the joint probability $P(X_1, \cdots, X_n \mid C)$

- Naïve Bayes classification
  - Assume that all input features are conditionally independent!

$$\begin{aligned} P(X_1, X_2, \cdots, X_n \mid C) &= P(X_1 \mid X_2, \cdots, X_n, C)P(X_2, \cdots, X_n \mid C) \\ &= P(X_1 \mid C)P(X_2, \cdots, X_n \mid C) \\ &= P(X_1 \mid C)P(X_2 \mid C) \cdots P(X_n \mid C) \end{aligned}$$

  - MAP classification rule: for $\mathbf{x} = (x_1, x_2, \cdots, x_n)$

$$[P(x_1 \mid c^*) \cdots P(x_n \mid c^*)]P(c^*) > [P(x_1 \mid c) \cdots P(x_n \mid c)]P(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

Naïve Bayes Classifier (cont.)

- Algorithm: Discrete-Valued Features
  - Learning Phase: Given a training set **S** of F features and L classes,

    For each target value of c_i $(c_i = c_1, \cdots, c_L)$

    $\hat{P}(C = c_i) \leftarrow$ estimate $P(C = c_i)$ with examples in **S**;

    For every feature value x_{jk} of each feature X_j $(j = 1, \cdots, F; k = 1, \cdots, N_j)$

    $\hat{P}(X_j = x_{jk} \mid C = c_i) \leftarrow$ estimate $P(X_j = x_{jk} \mid C = c_i)$ with examples in **S**;

    Output: $F * L$ conditional probabilistic (generative) models
  - Test Phase: Given an unknown instance $\mathbf{X}' = (a_1', \cdots, a_n')$

    "Look up tables" to assign the label c^* to **X'** if

$$[\hat{P}(a_1' \mid c^*) \cdots \hat{P}(a_n' \mid c^*)]\hat{P}(c^*) > [\hat{P}(a_1' \mid c) \cdots \hat{P}(a_n' \mid c)]\hat{P}(c), \quad c \neq c^*, c = c_1, \cdots, c_L$$

Example

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example (cont.)

- Learning Phase

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14$ $P(\text{Play}=No) = 5/14$

Example (cont.)

- Test Phase
  - **Given a new instance, predict its label**

    x'=(Outlook=*Sunny,* Temperature=*Cool,* Humidity=*High,* Wind=*Strong*)

  - **Look up tables achieved in the learning phrase**

P(Outlook=*Sunny*|Play=*Yes*) = 2/9
P(Temperature=*Cool*|Play=*Yes*) = 3/9
P(Huminity=*High*|Play=*Yes*) = 3/9
P(Wind=*Strong*|Play=*Yes*) = 3/9
P(Play=*Yes*) = 9/14

P(Outlook=S*unny*|Play=*No*) = 3/5
P(Temperature=*Cool*|Play==*No*) = 1/5
P(Huminity=*High*|Play=*No*) = 4/5
P(Wind=*Strong*|Play=*No*) = 3/5
P(Play=*No*) = 5/14

  - **Decision making with the MAP rule**

$P(Yes|\mathbf{x}') \approx [P(Sunny|Yes)P(Cool|Yes)P(High|Yes)P(Strong|Yes)]P(Play=Yes) = 0.0053$

$P(No|\mathbf{x}') \approx [P(Sunny|No)\ P(Cool|No)P(High|No)P(Strong|No)]P(Play=No) = 0.0206$

Given the fact $P(Yes|\mathbf{x}') < P(No|\mathbf{x}')$, we label $\mathbf{x}'$ to be "*No*".

Logistic Regression

Linear Methods for Classification

- *Decision boundaries* are linear.

- Two class problem:
  - The decision boundary between the two classes is a *hyperplane* in the feature vector space.

  - A hyperplane in the *p* dimensional input space is the set:

$$\{x : \alpha_o + \sum_{j=1}^{p} \alpha_j x_j = 0\}$$

Linear Methods for Classification (cont.)

- The two regions separated by a hyperplane:

$$\{x : \alpha_o + \sum_{j=1}^{p} \alpha_j x_j > 0\} \qquad \{x : \alpha_o + \sum_{j=1}^{p} \alpha_j x_j < 0\}$$

- For more than two classes, the decision boundary between any pair of classes *k* and *l* is a hyperplane.

- Example method for deciding the hyperplane:
  - Logistic regression

Logistic Regression

- There are two big branches of methods for classification. One is called *generative* modeling (which we saw with Naïve Bayes classification), and the other is called *discriminative* modeling.
  - A **generative** model learns the joint probability distribution.
  - A **discriminative** model learns the conditional probability distribution.

- *Logistic regression* for classification is a discriminative modeling approach, where we estimate the *posterior probabilities* of classes given *X* directly without assuming the marginal distribution on *X*.

- As a result, this method preserves linear classification boundaries.

Logistic Regression (cont.)

- A review of Bayes rule showed us that when we use 0-1 loss, we pick the class *k* that has the maximum posterior probability:

$$\hat{G}(x) = \arg\max_k Pr(G = k|X = x)$$

- The decision boundary between classes *k* and *l* is determined by:

$$Pr(G = k|X = x) = Pr(G = l|X = x)$$

- That is, the *x*'s at which the two posterior probabilities of *k* and *l* are equal.

Logistic Regression (cont.)

- If we divide both sides by $Pr(G = l \mid X = x)$ and take the log of this ratio, the previous equation is equivalent to:

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = 0$$

- Since we want to enforce a linear classification boundary, we assume the function above is linear:

$$\log \frac{Pr(G = k|X = x)}{Pr(G = l|X = x)} = a_0^{(k,l)} + \sum_{j=1}^{p} a_j^{(k,l)} x_j$$

- This is the basic assumption of logistic regression.

Logistic Regression (cont.)

- The log ratios of posterior probabilities are called *log-odds* or *logit* transformations.

- The posterior probabilities are given by the following two equations:

$$Pr(G=k|X=x)=\frac{\exp\left(\beta_{k0}+\beta_k^T x\right)}{1+\sum_{l=1}^{K-1}\exp\left(\beta_{l0}+\beta_l^T x\right)} \text{ for } k=1,\ldots,K-1$$

$$Pr(G=K|X=x)=\frac{1}{1+\sum_{l=1}^{K-1}\exp\left(\beta_{l0}+\beta_l^T x\right)}$$

Logistic Regression (cont.)

- For Pr($G = k \mid X = x$) given previously:
  - These must sum to 1: $\sum_{k=1}^{K} Pr(G = k|X = x) = 1$

- Similarities with linear regression on indicators:
  - Both attempt to estimate Pr($G = k \mid X = x$), both have linear classification boundaries, and the posterior probabilities sum to 1 across classes

- Differences with linear regression on indicators:
  - For linear regression, approximate Pr($G = k \mid X = x$) by a linear function of x; it is not guaranteed to fall between 0 and 1.
  - For logistic regression, Pr($G = k \mid X = x$) is a nonlinear (sigmoid) function of x; it is guaranteed to range from 0 to 1.

Logistic Regression (cont.)

- How do we estimate the parameters and how do we fit a logistic regression model?

- What we want to do is find parameters that *maximize* the conditional *likelihood* of class labels *G* given *X* using the training data.

- We use the **Newton Raphson Algorithm** (a gradient descent method).

Logistic Regression (cont.)

- Starting with β^{old}, a single Newton-Raphson update is given by this matrix formula:

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial l(\beta)}{\partial \beta}$$

where the derivatives are evaluated at β^{old}.

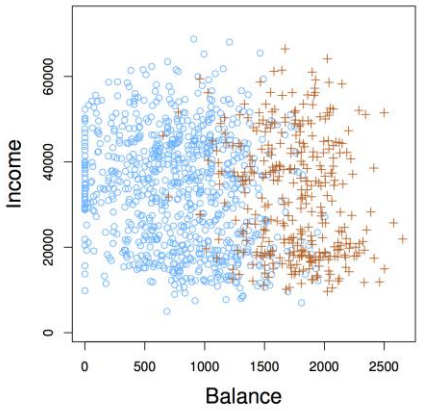
- If given an old set of parameters, we update the new set of parameters by taking β^{old} minus the inverse of the Hessian matrix times the first-order derivative vector.

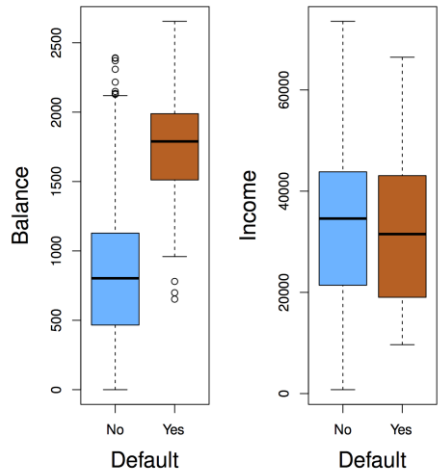
Logistic Regression: Credit Card Default Example

- We would like to be able to predict customers that are likely to default on their credit card.

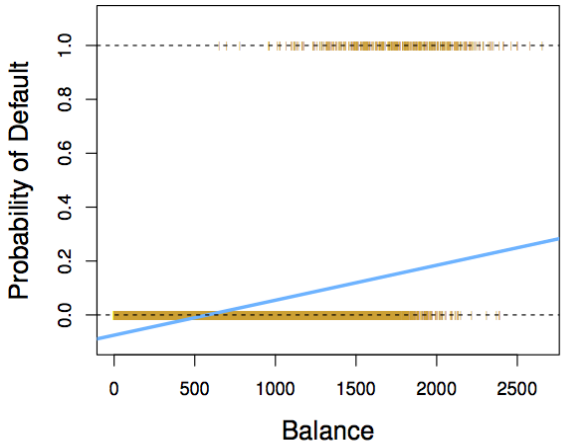
- Possible *X* variables:
  - Annual Income
  - Monthly Credit Card Balance

- The *Y* variable (Default) is categorical: Yes (1) or No (0)

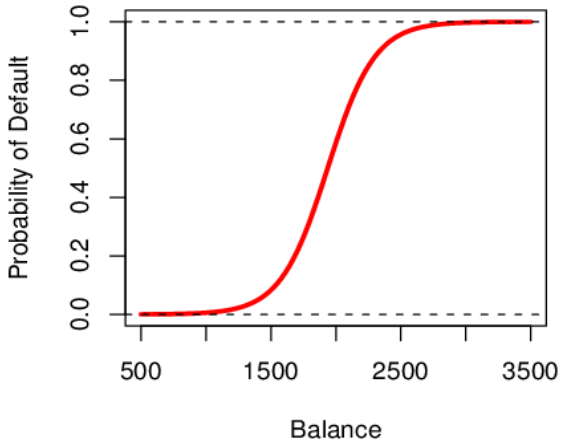




Logistic Regression: Credit Card Default Example (cont.)



- If we fit a linear regression model to the Default data, then:
  - For very low balances we predict a negative probability!
  - For high balances we predict a probability above 1!



- If we fit a logistic regression model, then the probability of default is between 0 and 1 for all balances.

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

Logistic Regression: Credit Card Default Example (cont.)

- Interpreting what β_1 means is not very easy with logistic regression, simply because we are predicting Pr(*Y*) and not *Y*.

- In a logistic regression model, increasing *X* by one unit changes the *log odds* by β_1, or equivalently it multiplies the odds by e^{β_1}.

- If $\beta_1 = 0$, then there is <u>no</u> relationship between *Y* and *X*. If $\beta_1 > 0$, then when *X* gets larger so does the probability that $Y = 1$. If $\beta_1 < 0$, then when *X* gets larger the probability that $Y = 1$ gets smaller.

- How much increase or decrease in probability depends on the slope.

Logistic Regression: Credit Card Default Example (cont.)

- We still want to perform a hypothesis test to see whether we can be sure that β_0 and β_1 are significantly different from zero.

- We use a Z test and interpret the p-value as usual.

- In this example, the p-value for balance is very small and $\hat{\beta}_1$ is positive. This means that if the balance increases, then the probability of default will increase as well.

	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

Logistic Regression: Credit Card Default Example (cont.)

	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

- What is the estimated probability of default for someone with a balance of \$1000?

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

- What is the estimated probability of default for someone with a balance of \$2000?

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

Logistic Regression: Credit Card Default Example (cont.)

- We can also use student (0,1) as a predictor to estimate the probability that an individual defaults:

	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

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) =$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) =$$

QUESTIONS????