S&DS 265 / 565
Introductory Machine Learning

Classification

September 15

Yale

Notes

- Assn 1 is posted; due in 2 weeks (midnight)
- Please join us at office hours!
- Use Ed Discussion for questions
- Some notes at an appropriate level:

  ▶ Background concepts:
    http://www.mit.edu/~6.s085/notes/lecture1.pdf

  ▶ Linear regression:
    http://www.mit.edu/~6.s085/notes/lecture3.pdf

Outline—Next two classes

- Some important concepts

- Logistic regression

- Generative vs. discriminative

- Gaussian discriminant analysis

- Examples: Supernovae and political blogs

- Regularization

- Algorithms for fitting the models

Working example: Fisher's Iris data

Outline—today

- Some important concepts
- Logistic regression
- Examples in Jupyter: Mushrooms and flowers
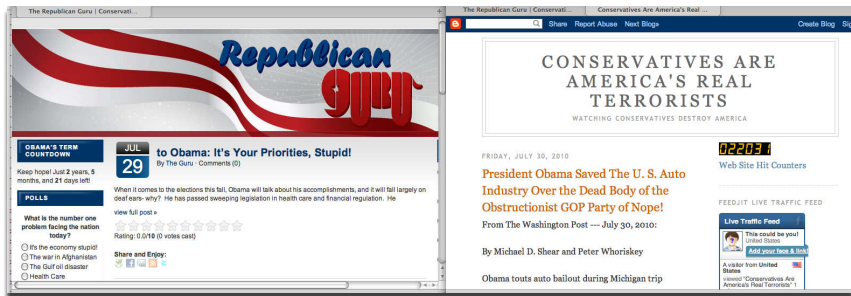
Classification tasks

- The Coronary Risk-Factor Study (CORIS). Data: 462 males between ages of 15 and 64 from three rural areas in South Africa.

  Outcome Y is presence ($Y = 1$) or absence ($Y = 0$) of coronary heart disease

  9 covariates: systolic blood pressure, cumulative tobacco (kg), ldl (low density lipoprotein cholesterol), adiposity, famhist (family history of heart disease), typea (type-A behavior), obesity, alcohol (current alcohol consumption), and age.
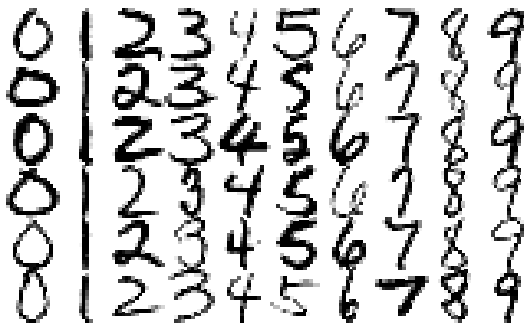
Classification tasks

- Political Blog Classification. A collection of 403 political blogs were collected during two months before the 2004 presidential election. The goal is to predict whether a blog is *liberal* ($Y = 0$) or *conservative* ($Y = 1$) given the content of the blog.
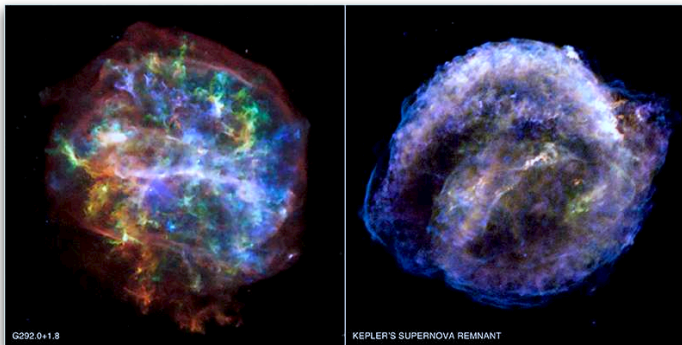
Classification tasks

- Handwriting Digit Recognition. Here each Y is one of the ten digits from 0 to 9. There are 256 covariates $X_1, \ldots, X_{256}$ corresponding to the intensity values of the pixels in a 16×16 image.

Classification tasks

- A supernova is an exploding star. Type Ia supernovae are a special class of supernovae that are very useful in astrophysics research. These supernovae have a characteristic *light curve*, which is a plot of the luminosity of the supernova versus time.

Classification tasks

- Ad click-through prediction. Predict whether or not a user will click on an ad presented. Used for ranking ads and setting prices.

Classification tasks

- The Iris Flower study. The data are 50 samples from each of three species of Iris flowers, *Iris setosa*, *Iris virginica* and *Iris versicolor*. The length and width of the sepal and petal are measured for each specimen based on these features.

- App for wildflowers

Iris setosa (Left), *Iris versicolor* (Middle), and *Iris virginica* (Right).

Fisher's iris classification

Iris setosa (Left), *Iris versicolor* (Middle), and *Iris virginica* (Right).
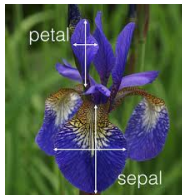
Important concepts

Binary classifier h: function from $\mathcal{X}$ to $\{0, 1\}$.

Linear if exists a function $H(x) = \beta_0 + \beta^T x$ such that
$h(x) = 1$ if $H(x) > 0$; 0 otherwise.

$H(x)$ also called a *linear discriminant function*. Decision boundary:
set $\{x \in \mathbb{R}^d : H(x) = 0\}$

Important concepts

Classification risk, or *error rate*, of *h*:

$$R(h) = \mathbb{P}(Y \neq h(X))$$

and the *empirical classification error* or *training error* is

$$\widehat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} I(h(x_i) \neq y_i).$$

Optimal classification rule

The optimal rule h^* is called the *Bayes rule*.

The risk $R^* = R(h^*)$ of the Bayes rule is called the *Bayes risk*.

The set $\{x \in \mathcal{X} : m(x) = 1/2\}$ is called the *Bayes decision boundary*.

The Bayes decision rule

Recall Bayes' rule (theorem):

$$\mathbb{P}(A \mid B) = \frac{\mathbb{P}(B \mid A)\,\mathbb{P}(A)}{\mathbb{P}(B)}$$

The Bayes decision rule

From Bayes' theorem

$$\mathbb{P}(Y = 1 \mid X = x) = \frac{\mathbb{P}(X = x \mid Y = 1)\,\mathbb{P}(Y = 1)}{\mathbb{P}(X = x)}$$

$$= \frac{p(x \mid Y = 1)\mathbb{P}(Y = 1)}{p(x \mid Y = 1)\mathbb{P}(Y = 1) + p(x \mid Y = 0)\mathbb{P}(Y = 0)}$$

$$= \frac{\pi_1 p_1(x)}{\pi_1 p_1(x) + (1 - \pi_1)p_0(x)}$$

where $\pi_1 = \mathbb{P}(Y = 1)$.

The Bayes decision rule

T he Bayes decision rule is then

$$\frac{p_1(x)}{p_0(x)} > \frac{1 - \pi_1}{\pi_1}.$$

Can be rewritten as

$$h^*(x) = \begin{cases} 1 & \text{if } \dfrac{p_1(x)}{p_0(x)} > \dfrac{1 - \pi_1}{\pi_1} \\ 0 & \text{otherwise.} \end{cases}$$

Small dataset example
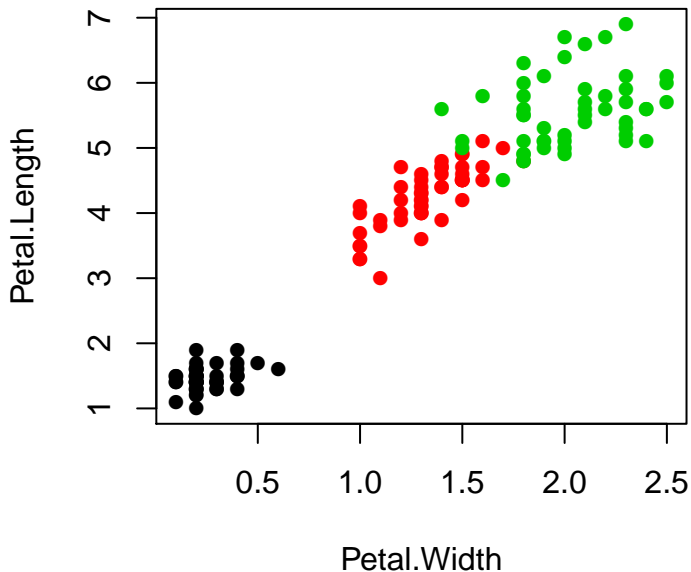
Simulated data–two classes
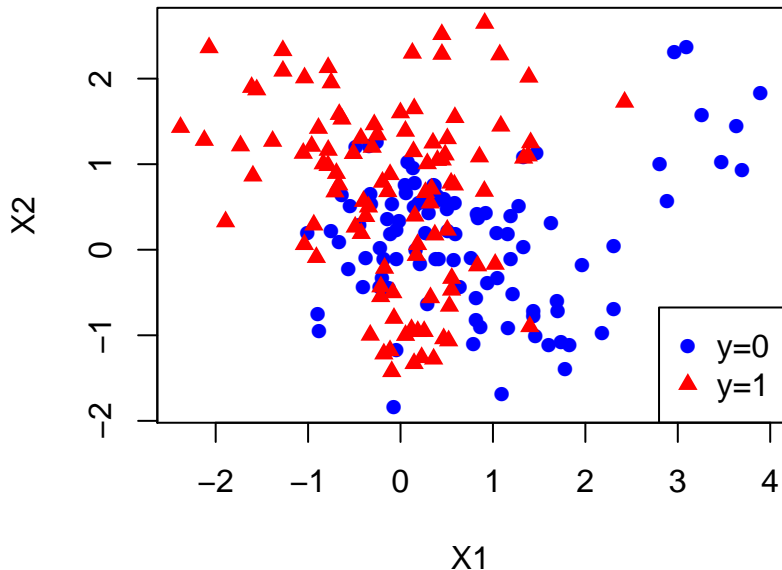
Simplification–one predictor
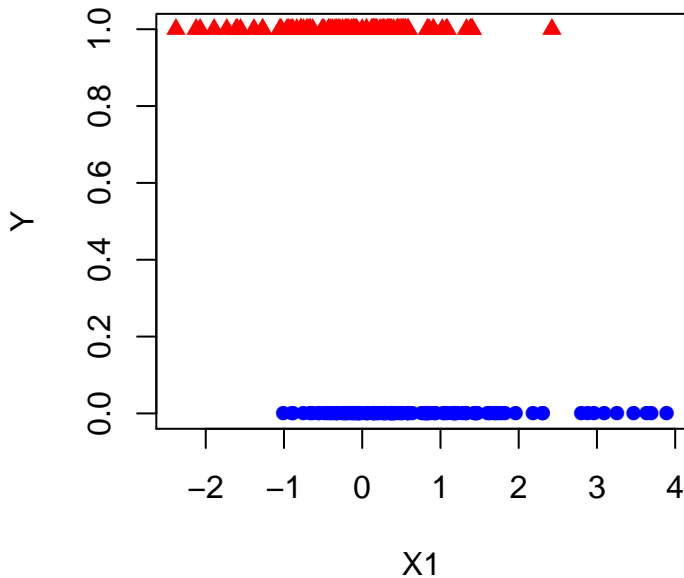
Logistic regression (binary case)

Conditional probabilities of the class:

$$\mathbb{P}(Y_i = 1 \mid X = x_i) = p(x_i)$$

$$\mathbb{P}(Y_i = 0 \mid X = x_i) = 1 - p(x_i)$$

Logistic regression (binary case)

Conditional probabilities of the class:

$$\mathbb{P}(Y_i = 1 \mid X = x_i) = p(x_i)$$

$$\mathbb{P}(Y_i = 0 \mid X = x_i) = 1 - p(x_i)$$

We model the relationship between $p(x_i)$ and x_i.

Logistic regression
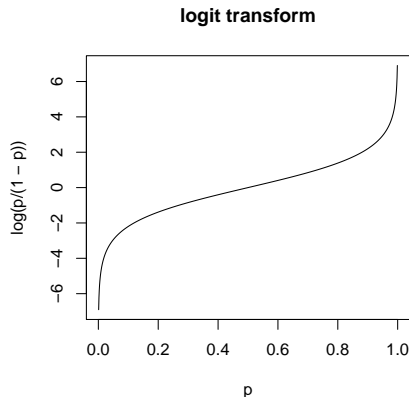
logit transform

The *logit* transform:

$$logit(p) = \log\left(\frac{p}{1-p}\right)$$

Logistic regression
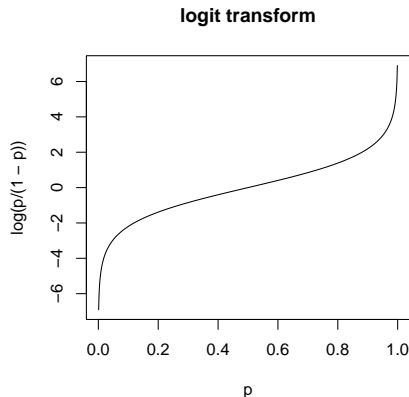
logit transform

The *logit* transform:

$$logit(p) = \log\left(\frac{p}{1-p}\right)$$

The logit transform

- is monotone
- maps the interval $[0, 1]$ to $(-\infty, \infty)$

Logistic regression

Logistic regression is a linear regression model of the log odds:

$$logit(\widehat{p}(x)) = \widehat{\beta}_0 + \widehat{\beta}_1 x$$

- p is a probability.
- $\frac{p}{1-p}$ is odds.
- $logit(p) = \log\left(\frac{p}{1-p}\right)$ is (natural) log odds.

Logistic regression

Logistic regression is a linear regression model of the log odds:

$$logit(\widehat{p}(x)) = \widehat{\beta}_0 + \widehat{\beta}_1 x$$

- p is a probability.
- $\frac{p}{1-p}$ is odds.
- $logit(p) = \log\left(\frac{p}{1-p}\right)$ is (natural) log odds.

Equivalent formulation:

$$\widehat{p}(x) = \frac{e^{\widehat{\beta}_0 + \widehat{\beta}_1 x}}{1 + e^{\widehat{\beta}_0 + \widehat{\beta}_1 x}} = logistic(x^T \widehat{\beta}) \equiv softmax(x^T \widehat{\beta})$$

LR decision boundary is linear

- When $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$, $\frac{\widehat{p}}{1-\widehat{p}} = 1$, so $\widehat{p} = 0.5$.

LR decision boundary is linear

- When $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$, $\frac{\widehat{p}}{1-\widehat{p}} = 1$, so $\widehat{p} = 0.5$.

- If our goal is to minimize the overall training error rate, then we use the rule:
$$\widehat{y} = \begin{cases} 1 & \widehat{p} \geq 0.5 \\ 0 & \widehat{p} < 0.5 \end{cases}$$

LR decision boundary is linear

- When $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$, $\frac{\widehat{p}}{1-\widehat{p}} = 1$, so $\widehat{p} = 0.5$.

- If our goal is to minimize the overall training error rate, then we use the rule:
$$\widehat{y} = \begin{cases} 1 & \widehat{p} \geq 0.5 \\ 0 & \widehat{p} < 0.5 \end{cases}$$

- Hence, the decision boundary is given by $\{x : x^T \widehat{\beta} = 0\}$.

LR decision boundary is linear

- When $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$, $\frac{\widehat{p}}{1-\widehat{p}} = 1$, so $\widehat{p} = 0.5$.

- If our goal is to minimize the overall training error rate, then we use the rule:
$$\widehat{y} = \begin{cases} 1 & \widehat{p} \geq 0.5 \\ 0 & \widehat{p} < 0.5 \end{cases}$$

- Hence, the decision boundary is given by $\{x : x^T\widehat{\beta} = 0\}$.
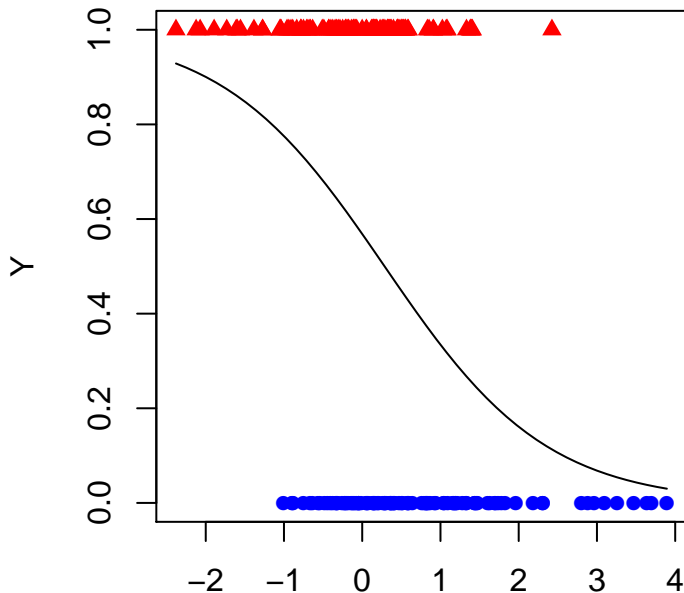
The decision boundary is linear in x!

Simulated data

Fitting a logistic regression

Traditionally, use maximum likelihood estimation (MLE).

- Likelihood of a single observation (x_i, y_i):

$$L_i(\beta) = p_i^{y_i} \cdot (1 - p_i)^{1-y_i} = \left(\frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \right)^{y_i} \cdot \left(1 - \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \right)^{1-y_i}$$

Fitting a logistic regression

Traditionally, use maximum likelihood estimation (MLE).

- Likelihood of a single observation (x_i, y_i):

$$L_i(\beta) = p_i^{y_i} \cdot (1 - p_i)^{1-y_i} = \left(\frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \right)^{y_i} \cdot \left(1 - \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \right)^{1-y_i}$$

- Log-likelihood of a single observation:

$$\ell_i(\beta) = y_i \log \left(\frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \right) + (1 - y_i) \log \left(\frac{1}{1 + e^{x_i^T \beta}} \right)$$

$$= y_i x_i^T \beta - \log(1 + e^{x_i^T \beta})$$

Fitting a logistic regression

Traditionally, use maximum likelihood estimation (MLE).

- Likelihood of a single observation (x_i, y_i):

$$L_i(\beta) = p_i^{y_i} \cdot (1 - p_i)^{1-y_i} = \left(\frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \right)^{y_i} \cdot \left(1 - \frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \right)^{1-y_i}$$

- Log-likelihood of a single observation:

$$\ell_i(\beta) = y_i \log \left(\frac{e^{x_i^T \beta}}{1 + e^{x_i^T \beta}} \right) + (1 - y_i) \log \left(\frac{1}{1 + e^{x_i^T \beta}} \right)$$

$$= y_i x_i^T \beta - \log(1 + e^{x_i^T \beta})$$

- Sum this over all data points

Extension to more than 2 classes

Multinomial logistic regression extends the logistic regression model to $K \geq 2$ classes.

$$\log \left(\frac{P(Y = k \mid X = x)}{P(Y = 0 \mid X = x)} \right) = x^T \beta_k, \qquad k = 1, 2, \ldots, K - 1$$

Extension to more than 2 classes

Multinomial logistic regression extends the logistic regression model to $K \geq 2$ classes.

$$\log\left(\frac{P(Y = k \mid X = x)}{P(Y = 0 \mid X = x)}\right) = x^T\beta_k, \qquad k = 1, 2, \ldots, K-1$$

$$P(Y = k \mid X = x) = \frac{\exp(x^T\beta_k)}{1 + \sum_{l=1}^{K-1}\exp(x^T\beta_l)}, \qquad k = 1, 2, \ldots, K-1$$

Extension to more than 2 classes

Multinomial logistic regression extends the logistic regression model to $K \geq 2$ classes.

$$\log \left(\frac{P(Y = k \mid X = x)}{P(Y = 0 \mid X = x)} \right) = x^T \beta_k, \qquad k = 1, 2, \ldots, K - 1$$
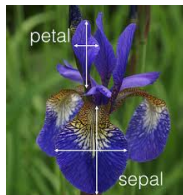
$$P(Y = k \mid X = x) = \frac{\exp(x^T \beta_k)}{1 + \sum_{l=1}^{K-1} \exp(x^T \beta_l)}, \qquad k = 1, 2, \ldots, K - 1$$

$$P(Y = \cdot \mid X = x) = \text{softmax} \left(1, \exp(x^T \beta_1), \ldots, \exp(x^T \beta_{K-1}) \right)$$
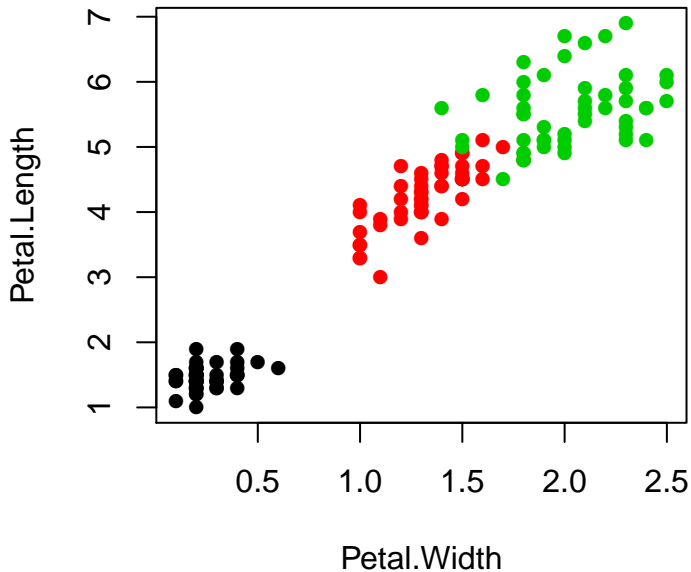
Fisher's iris classification

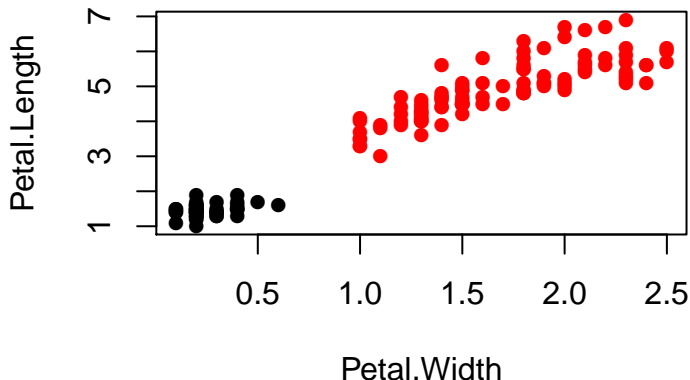*Iris setosa* (Left), *Iris versicolor* (Middle), and *Iris virginica* (Right).
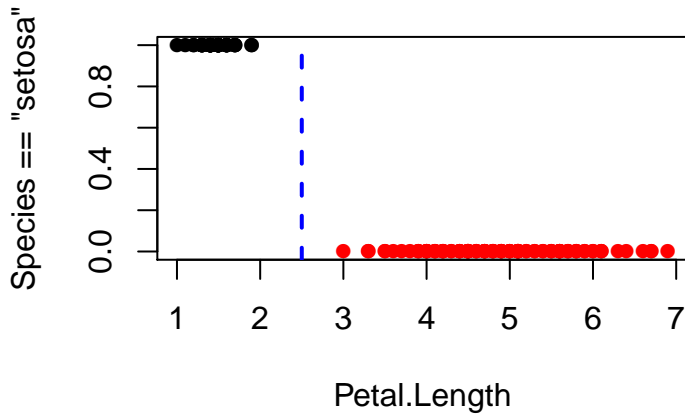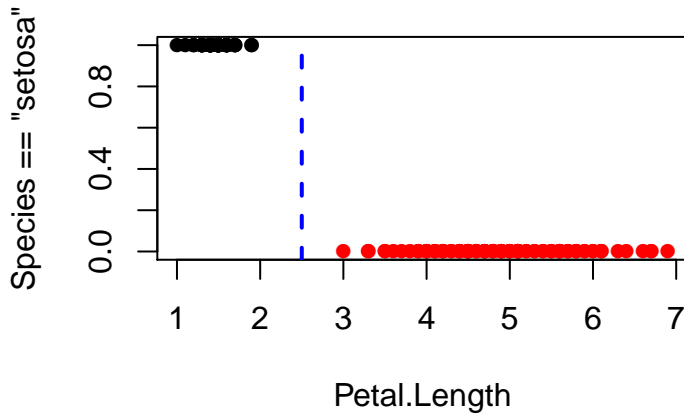
Separable classes

Separable classes

Pretend we only care for predicting setosas ($Y = 1$) vs. non-setosas ($Y = 0$):

Separable classes

Separable classes

Petal length of 2.5 can perfectly separate $Y = 1$ and $Y = 0$ groups.
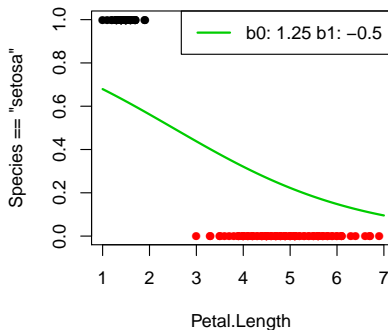
Separable classes

Decision boundary: $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$.

$\widehat{\beta}_1 = -\frac{\widehat{\beta}_0}{2.5}$ for $\widehat{\beta}_1 < 0$ will yield perfect fits.

Separable classes

Decision boundary: $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$.
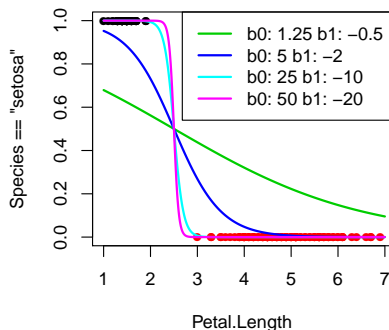
$\widehat{\beta}_1 = -\frac{\widehat{\beta}_0}{2.5}$ for $\widehat{\beta}_1 < 0$ will yield perfect fits.

Separable classes

Decision boundary: $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$.

$\widehat{\beta}_1 = -\frac{\widehat{\beta}_0}{2.5}$ for $\widehat{\beta}_1 < 0$ will yield perfect fits.

Int	Slope	Likelihood
1.25	-0.5	0.0000000
5.00	-2.0	0.0001696
25.00	-10.0	0.9846004
50.00	-20.0	0.9999415

Separable classes

Decision boundary: $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$.

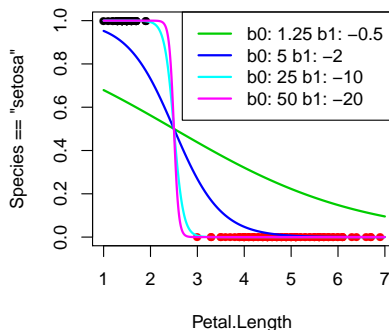$\widehat{\beta}_1 = -\frac{\widehat{\beta}_0}{2.5}$ for $\widehat{\beta}_1 < 0$ will yield perfect fits.

Int	Slope	Likelihood
1.25	-0.5	0.0000000
5.00	-2.0	0.0001696
25.00	-10.0	0.9846004
50.00	-20.0	0.9999415

As $\|\beta\|$ increases, likelihood approaches 1.

Separable classes

Problematic?

- Appears that predictor is not informative, but it is!

Separable classes

Problematic?

- Appears that predictor is not informative, but it is!
- Theoretically we obtained a perfect fit on the training data.

Separable classes

Problematic?

- Appears that predictor is not informative, but it is!
- Theoretically we obtained a perfect fit on the training data.
  - ▶ Overfitting is possible. Regularization can help.

Examples in Jupyter notebook

Lets work through some examples in a Jupyter notebook. Please open `classification-examples.ipynb` and run the notebook as we go through it.

Summary

- In classification we predict a class label

- The default model is logistic regression — corresponds to linear regression

- The model is fit to maximize the probability of the data

- If the data are linearly separable, this causes numerical problems

- One parameter for each input variable — later we will discuss neural nets and other methods to learn good features of the input