

S&DS 265 / 565  
Introductory Machine Learning







# **Classification (continued)**

September 21

# Upcoming items

- Assn 1 due on Thursday (midnight, 11:59pm)
- Assn 2 will be posted on Thursday
- Submit *both* your `.ipynb` notebook and a printout as `.pdf` (save as HTML then print as pdf).
- Quiz 1 will be available on Canvas for 24 hours starting at noon; you have 20 minutes to take the quiz.
- Questions?

# Updates to Calendar `iml.ydata123.org`

Week	Dates	Topics	Demos	Assignments & Exams	Slides and Readings
1	Sept 2	Course overview			Sept 2: <a href="#">Course overview</a>
2	Sept 7, 9	Python and background concepts	 <a href="#">Python elements</a>  <a href="#">Covid trends</a>	Thu: <a href="#">Quiz 0</a>	<a href="#">Data8 Chapters 3, 4, 5</a> Sept 7: <a href="#">Python elements</a> Sept 9: <a href="#">Pandas and linear regression</a>
3	Sept 14, 16	Linear regression and classification	 <a href="#">Covid trends (revisited)</a>  <a href="#">Classification examples</a>	Tue:  <a href="#">Assn1 out</a>	Sept 14: <a href="#">Regression concepts</a> <a href="#">Notes on regression</a> Sept 16: <a href="#">Classification</a> <a href="#">Notes on classification</a>
4	Sept 21, 23	Stochastic gradient descent	 <a href="#">SGD examples</a>	Tue: <a href="#">Quiz 1</a> Thu: <a href="#">Assn 1 in</a> ; <a href="#">Assn 2 out</a>	Sept 21: <a href="#">Classification (continued)</a>

Many parts of the notes are



# Outline

- Logistic regression (continued)
- Iris example
- Generative vs. discriminative
- Gaussian discriminant analysis
- Regularization
- Example: Supernovae
- Next: Algorithms for fitting the models

# Recall: 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  $\{x \in \mathbb{R}^d : H(x) = 0\}$

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

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

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

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

# Optimal classification rule

**Theorem.** The classification rule that minimizes  $R(h)$  is

$$h^*(x) = \begin{cases} 1 & \text{if } m(x) > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

where  $m(x) = \mathbb{E}(Y | X = x) = \mathbb{P}(Y = 1 | X = x)$  denotes the regression function.

This 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

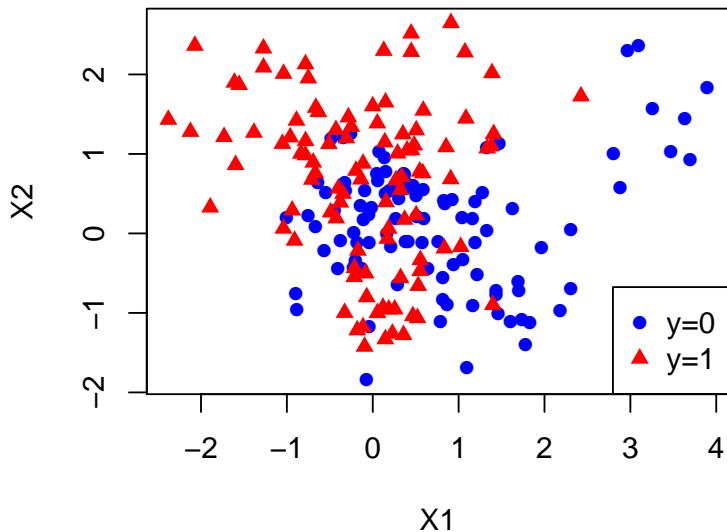
From Bayes' theorem

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

So,

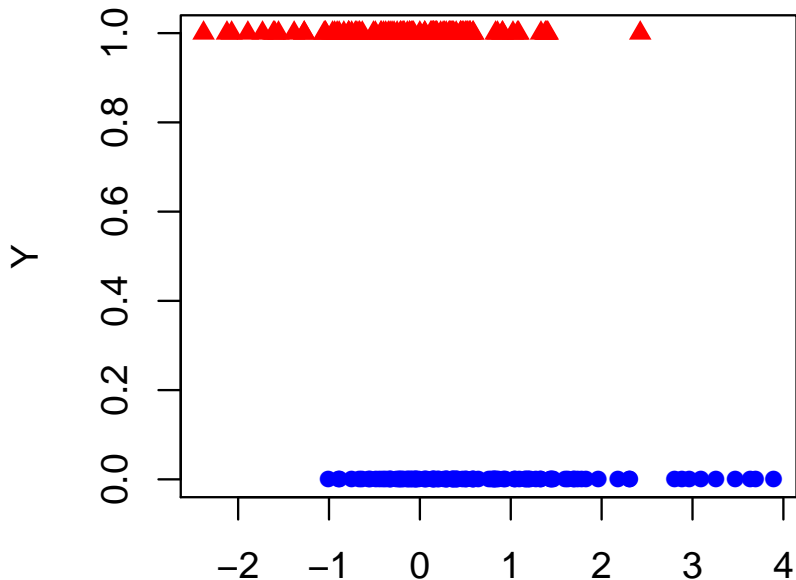
$$m(x) > \frac{1}{2} \text{ is equivalent to } \frac{p(x | Y = 1)}{p(x | Y = 0)} > \frac{\mathbb{P}(Y = 0)}{\mathbb{P}(Y = 1)}.$$

## Simulated data: Large Bayes error





## Simplification—one predictor: Large Bayes error



# Logistic regression

Conditional probabilities of the class:

$$\mathbb{P}(Y = 1 \mid X = x) \equiv p(x)$$

$$\mathbb{P}(Y = 0 \mid X = x) = 1 - p(x)$$

# Logistic regression

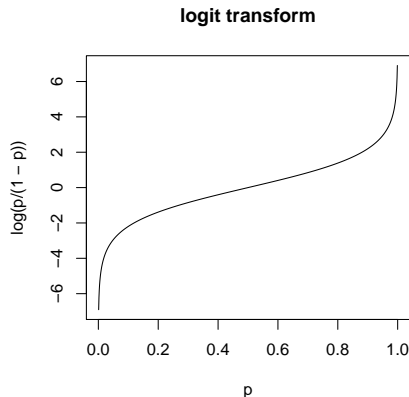
Conditional probabilities of the class:

$$\mathbb{P}(Y = 1 \mid X = x) \equiv p(x)$$

$$\mathbb{P}(Y = 0 \mid X = x) = 1 - p(x)$$

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

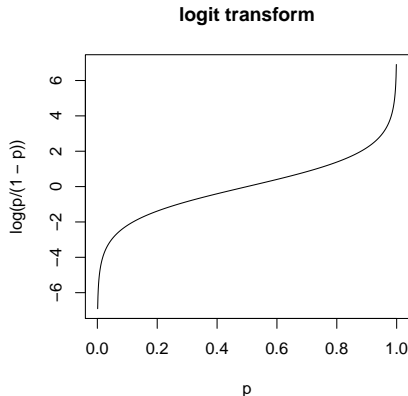
# Logistic regression



The *logit* transform:

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

# Logistic regression



The *logit* transform:

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

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:

$$\text{logit}(p(x)) = \beta_0 + \beta_1 x$$

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

# Logistic regression

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

$$\text{logit}(p(x)) = \beta_0 + \beta_1 x$$

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

Equivalent formulation:

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}} = \text{logistic}(x^T \beta) \equiv \text{softmax}(x^T \beta)$$

# LR decision boundary is linear

- When  $\hat{\beta}_0 + \hat{\beta}_1 x = 0$ ,  $\frac{\hat{p}}{1-\hat{p}} = 1$ , so  $\hat{p} = \frac{1}{2}$ .



# LR decision boundary is linear

- When  $\hat{\beta}_0 + \hat{\beta}_1 x = 0$ ,  $\frac{\hat{p}}{1-\hat{p}} = 1$ , so  $\hat{p} = \frac{1}{2}$ .
- If our goal is to minimize the overall training error rate, then we use the rule:

$$\hat{y} = \begin{cases} 1 & \hat{p} > \frac{1}{2} \\ 0 & \hat{p} \leq \frac{1}{2} \end{cases}$$

# LR decision boundary is linear

- When  $\hat{\beta}_0 + \hat{\beta}_1 x = 0$ ,  $\frac{\hat{p}}{1-\hat{p}} = 1$ , so  $\hat{p} = \frac{1}{2}$ .
- If our goal is to minimize the overall training error rate, then we use the rule:

$$\hat{y} = \begin{cases} 1 & \hat{p} > \frac{1}{2} \\ 0 & \hat{p} \leq \frac{1}{2} \end{cases}$$

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

# LR decision boundary is linear

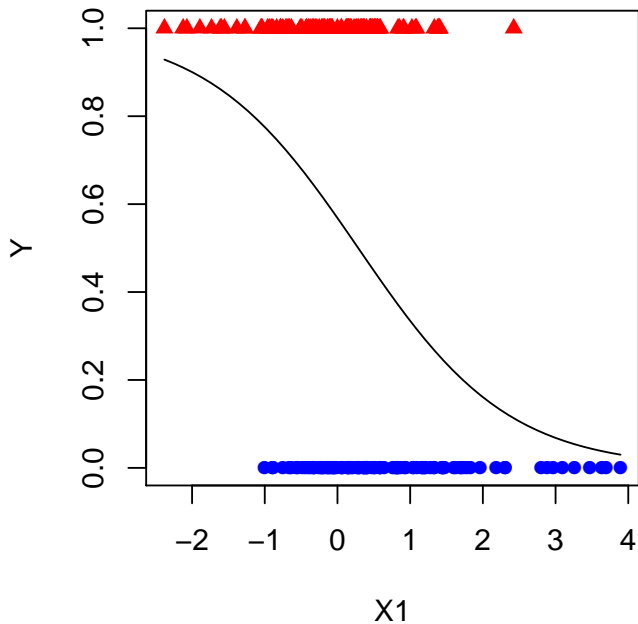
- When  $\hat{\beta}_0 + \hat{\beta}_1 x = 0$ ,  $\frac{\hat{p}}{1-\hat{p}} = 1$ , so  $\hat{p} = \frac{1}{2}$ .
- If our goal is to minimize the overall training error rate, then we use the rule:

$$\hat{y} = \begin{cases} 1 & \hat{p} > \frac{1}{2} \\ 0 & \hat{p} \leq \frac{1}{2} \end{cases}$$

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

The decision boundary is linear in  $x$ !

# Simulated data



# Fitting a logistic regression

Traditionally, use maximum likelihood estimation (MLE).

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

$$L_i(\beta) = p_i^{y_i} \cdot (1 - p_i)^{1-y_i}$$

# Fitting a logistic regression

Traditionally, use maximum likelihood estimation (MLE).

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

$$L_i(\beta) = p_i^{y_i} \cdot (1 - p_i)^{1-y_i}$$

- Log-likelihood of a single observation:

$$\begin{aligned}\ell_i(\beta) &= y_i \log p_i + (1 - y_i) \log(1 - p_i) \\ &= y_i x_i^T \beta - \log(1 + e^{x_i^T \beta})\end{aligned}$$

# Fitting a logistic regression

Traditionally, use maximum likelihood estimation (MLE).

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

$$L_i(\beta) = p_i^{y_i} \cdot (1 - p_i)^{1-y_i}$$

- Log-likelihood of a single observation:

$$\begin{aligned}\ell_i(\beta) &= y_i \log p_i + (1 - y_i) \log(1 - p_i) \\ &= y_i x_i^T \beta - \log(1 + e^{x_i^T \beta})\end{aligned}$$

- Aggregate log-likelihood:

$$\ell(\beta) = \sum_{i=1}^n \left\{ y_i x_i^T \beta - \log(1 + e^{x_i^T \beta}) \right\}$$

# Extension to more than 2 classes

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

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



## Extension to more than 2 classes

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

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

$$P(Y = k | X = x) = \begin{cases} \frac{\exp(x^T \beta_k)}{1 + \sum_{l=1}^{K-1} \exp(x^T \beta_l)} & k = 1, 2, \dots, K - 1 \\ \frac{1}{1 + \sum_{l=1}^{K-1} \exp(x^T \beta_l)} & k = 0 \end{cases}$$

# Loss function for 3 classes

We want to maximize the likelihood of the data, which is equivalent to minimizing the log-likelihood:

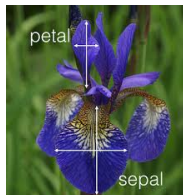
$$\begin{aligned} & - \sum_{i=1}^n \log P(Y = y_i | X = x_i) \\ &= - \sum_{i=1}^n \left\{ \beta_{y_i}^T x_i - \log(1 + e^{\beta_1^T x_i} + e^{\beta_2^T x_i}) \right\} \\ &= n \log(1 + e^{\beta_1^T x_i} + e^{\beta_2^T x_i}) - \sum_{i=1}^n \beta_{y_i}^T x_i \end{aligned}$$

keeping in mind that  $\beta_0$  is all zeros, by definition.

# Fisher's iris classification



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



# Examples in Jupyter notebook

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

# Regularization

Recall from last time: We can separate *setosa* from the two other species just on the basis of their petal length (or width).

This causes problems when we fit the model — the parameters get large so that the probabilities get very close to zero or one.

To address this problem, we *regularize* the parameters. This means we introduce a penalty term that prevents them from getting too large (in absolute value).

# Regularization: Simplest setting

The least squares estimator:

$$\hat{\beta} = \arg \min_{\beta} (y - \beta)^2$$

# Regularization: Simplest setting

The least squares estimator:

$$\hat{\beta} = \arg \min_{\beta} (y - \beta)^2$$

Solution:  $\hat{\beta} = y$

# Regularization: Simplest setting

The least squares estimator:

$$\hat{\beta} = \arg \min_{\beta} (y - \beta)^2$$

Solution:  $\hat{\beta} = y$

Now *penalize*  $\beta$  from getting too large:

$$\hat{\beta} = \arg \min_{\beta} (y - \beta)^2 + \lambda \beta^2$$



# Regularization: Simplest setting

The least squares estimator:

$$\hat{\beta} = \arg \min_{\beta} (y - \beta)^2$$

Solution:  $\hat{\beta} = y$

Now *penalize*  $\beta$  from getting too large:

$$\hat{\beta} = \arg \min_{\beta} (y - \beta)^2 + \lambda \beta^2$$

Solution:  $\hat{\beta} = \frac{y}{1+\lambda}$ .

# Regularization: Simplest setting

The least squares estimator:

$$\hat{\beta} = \arg \min_{\beta} (y - \beta)^2$$

Solution:  $\hat{\beta} = y$

Now *penalize*  $\beta$  from getting too large:

$$\hat{\beta} = \arg \min_{\beta} (y - \beta)^2 + \lambda \beta^2$$

Solution:  $\hat{\beta} = \frac{y}{1+\lambda}$ . As  $\lambda$  gets large,  $\hat{\beta}$  shrinks to zero.

# Two flavors of classifiers

*Generative models* model both the input  $X$  and the output  $Y$ .

*Discriminative models* model only the output  $Y$  given  $X$ .

# Two flavors of classifiers

*Generative models* model both the input  $X$  and the output  $Y$ .

*Discriminative models* model only the output  $Y$  given  $X$ .

*Which do you think is better?*

# Generative models

We build a model of the inputs  $x$  and the outputs  $y$

In the generative case we typically estimate the joint distribution by maximizing the *joint likelihood*:  $p(x, y)$

# Discriminative models

In the discriminative case we are concerned about the *conditional* distribution of the output given the input.

We will typically estimate by maximizing the *conditional likelihood*:

The form of the Bayes classification rule suggests we should use a generative model

$$m_{\theta}(x) \equiv \mathbb{P}(Y = 1 \mid X = x) = \frac{\pi_1 p_{\theta_1,1}(x)}{(1 - \pi_1) p_{\theta_0,0}(x) + \pi_1 p_{\theta_1,1}(x)}.$$

Given an estimator  $(\hat{\theta}_n, \hat{\pi}_1)$ , define classification rule

$$\hat{h}(x) = \mathbb{1}(m_{\hat{\theta}_n}(x) > 1/2).$$

where  $\mathbb{1}$  is the “indicator function” which is 1 if its argument is true, and zero otherwise.

# Gaussian discriminant analysis

- A type of generative model
- We model the inputs  $x$  using Gaussians
- Two flavors: Linear and Quadratic



# Quadratic discriminant analysis

In the binary (two-class) case, we have two Gaussians:

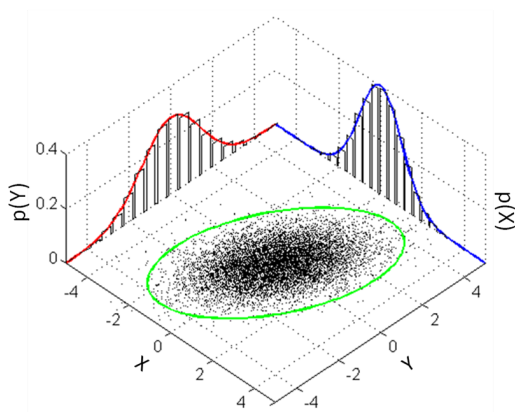
$$X \mid y = 1 \sim N(\mu_1, \Sigma_1)$$

$$X \mid y = 0 \sim N(\mu_0, \Sigma_0)$$

The decision boundary is a quadratic surface (algebra!)

# Quadratic discriminant analysis

To estimate this we just separate the training data according to the two labels and estimate two separate Gaussians. Easy-peasy!



---

Think of  $Y$  here as another predictor variable, not the class label!

[https://en.wikipedia.org/wiki/Multivariate\\_normal\\_distribution](https://en.wikipedia.org/wiki/Multivariate_normal_distribution)

# Linear discriminant analysis

In the binary (two-class) case, we again have two Gaussians:

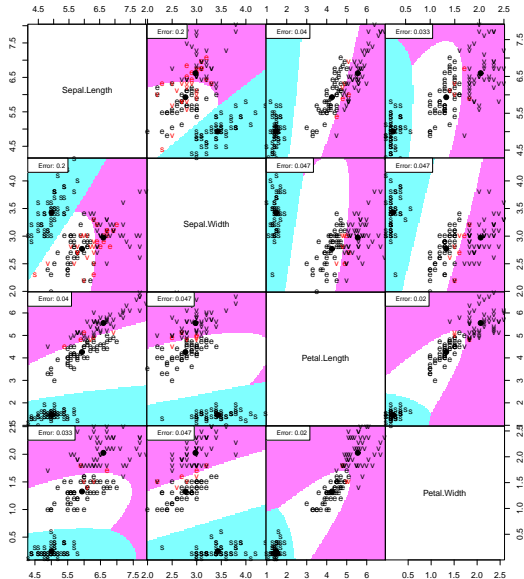
$$X \mid y = 1 \sim N(\mu_1, \Sigma)$$

$$X \mid y = 0 \sim N(\mu_0, \Sigma)$$

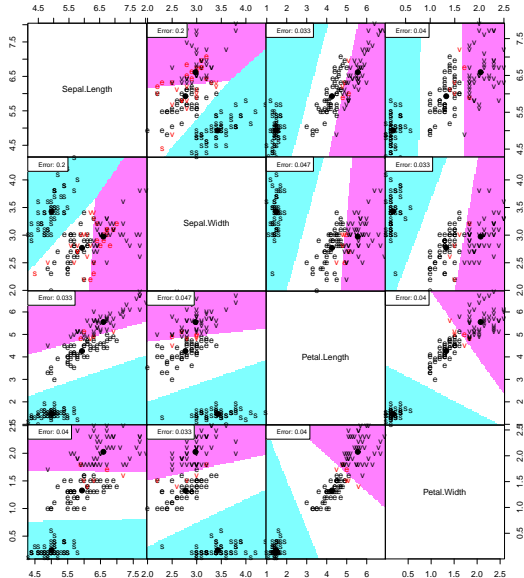
But now we use the same covariance matrix for each.

The decision boundary is now *linear*.

# Quadratic discriminant analysis: Iris data



# Linear discriminant analysis: Iris data



# Logistic regression

Logistic regression is a discriminative model, because we don't have a model for the inputs  $X$ .

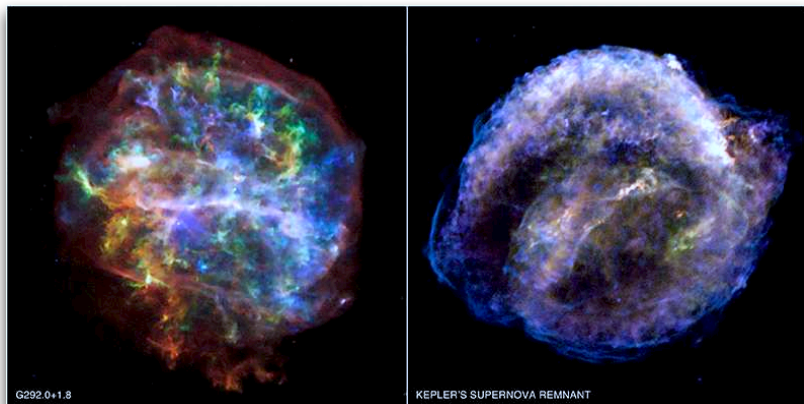
We only model the conditional probability  $p(Y | X)$ .

Logistic regression is the discriminative version of linear discriminative analysis (the latter is a generative model).

# Supernovæ

- A *supernova* is an exploding star.
- Type Ia supernovae are very useful in astrophysics research. Have a characteristic *light curve*, same maximum brightness
- Since we know both the absolute and apparent (measured) brightness of a type Ia supernova, we can compute its distance.
- Astronomers also measure the *redshift* of the supernova, the speed at which the supernova is moving away from us
- The relationship between distance and redshift provides important information about the large scale structure of the universe.

# Supernovae



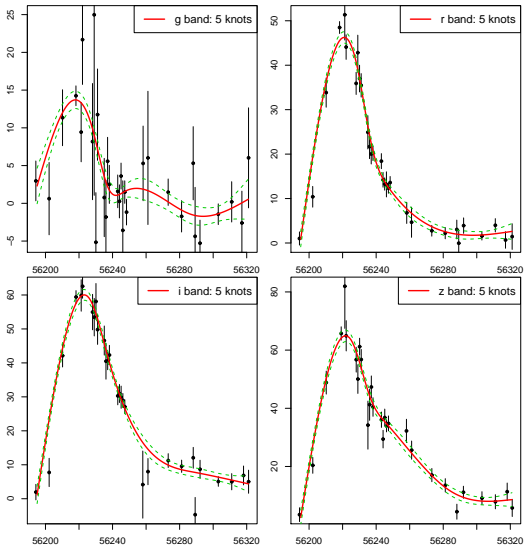
Two supernova remnants from the NASA's Chandra X-ray Observatory study. The right one is Type Ia. (Credit: NASA/CXC/UCSC/L. Lopez et al.)



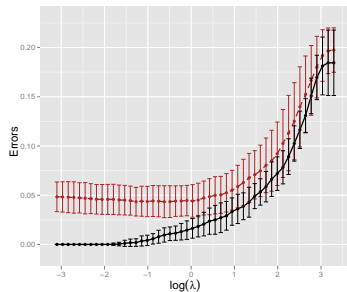
# Supernovae

- Data are 20,000 real and simulated supernovae.
- For each supernova, there are a few noisy measurements of the flux (brightness) in four different filters — *g*-band (green), *r*-band (red), *i*-band (infrared) and *z*-band (blue).
- These noisy data are processed to fit a curve through the measurements in each band.

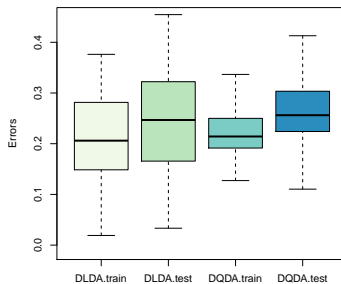
# Supernovae



# Supernovae – classification results



Logistic Regression



Discriminant Analysis

# Fitting a logistic regression model

- We maximize conditional likelihood. There is no closed form.
- Need to iterate.
- Standard approach is equivalent to Newton's algorithm
  - ▶ Make a quadratic approximation
  - ▶ Do a weighted least squares regression
  - ▶ Repeat

We'll talk about a more scalable approach next time

# Summary

- Classifiers come in two flavors: generative & discriminative
- Linear Gaussian discriminant analysis is a simple generative classifier
- Logistic regression is the discriminative version. Default method; no closed-form solution
- We regularize the parameters with a penalty  $\beta^2$  that keeps them from being too big. *Shrinks* coefficients toward zero.