S&DS 265 / 565

Introductory Machine Learning

Classification (continued)

September 21

Yale

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: Course overview
2	Sept 7, 9	Python and background concepts	CO Python elements CO Covid trends	Thu: Quiz 0	Data8 Chapters 3, 4, 5 Sept 7: Python elements Sept 9: Pandas and linear regression
3	Sept 14, 16	Linear regression and classification	CO Covid trends (revisited) CO Classification examples	Tue: CO Assn1 out	Sept 14: Regression concepts Notes on regression Sept 16: Classification Notes on classification
4	Sept 21, 23	Stochastic gradient descent	CO SGD examples	Tue: Quiz 1 Thu: Assn 1 in; Assn 2 out	Sept 21: Classification (continued)

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

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

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

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

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

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 $m(x) = \mathbb{E}(Y \mid X = x) = \mathbb{P}(Y = 1 \mid X = x)$ 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

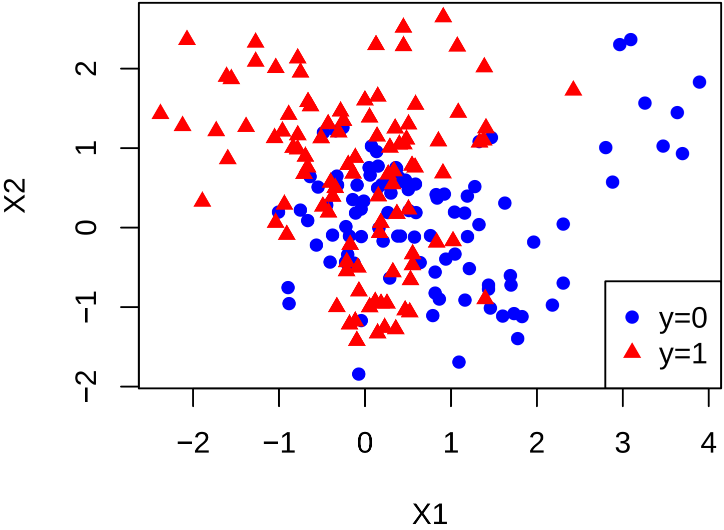
From Bayes' theorem

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

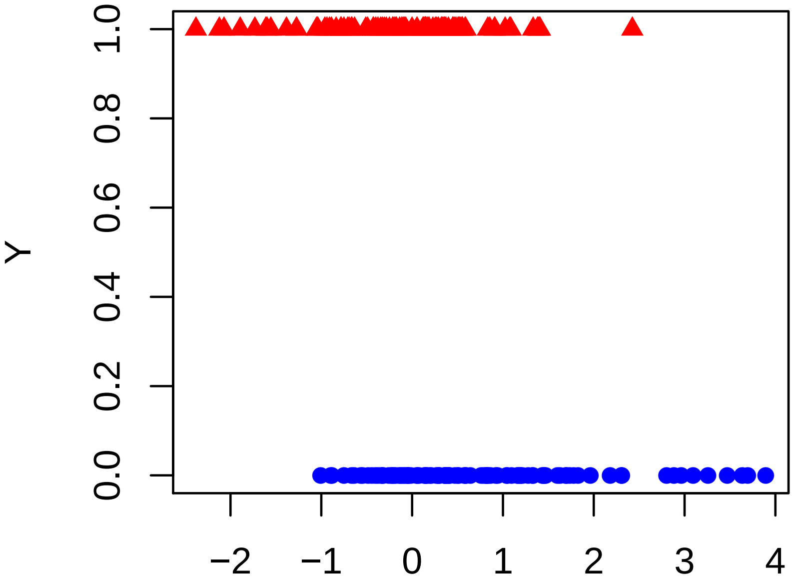
So,

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

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

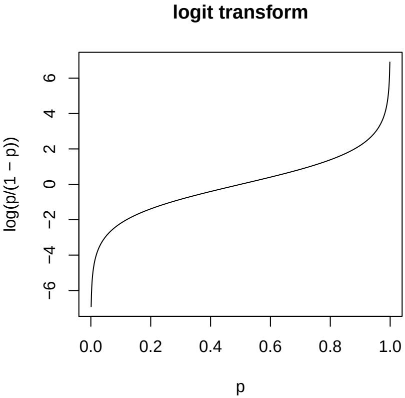
Conditional probabilities of the class:

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

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

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

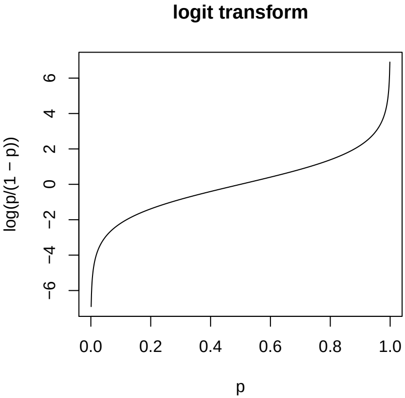
Logistic regression



The *logit* transform:

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

Logistic regression



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:

$$logit(p(x)) = \beta_0 + \beta_1 x$$

- 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:

$$logit(p(x)) = \beta_0 + \beta_1 x$$

- 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:

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

LR decision boundary is linear

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

LR decision boundary is linear

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

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

LR decision boundary is linear

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

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

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

LR decision boundary is linear

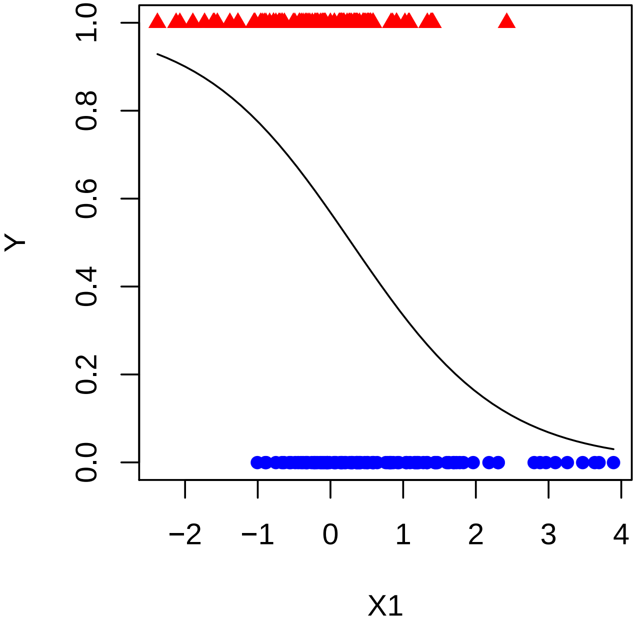
- When $\widehat{\beta}_0 + \widehat{\beta}_1 x = 0$, $\frac{\widehat{p}}{1-\widehat{p}} = 1$, so $\widehat{p} = \frac{1}{2}$.
- If our goal is to minimize the overall training error rate, then we use the rule:

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

- 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):

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

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.

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

Extension to more than 2 classes

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

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

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

Loss function for 3 classes

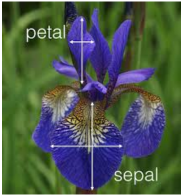
We want to maximize the likelihood of the data, which is equivalent to minimizing the log-likelihood:

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

keeping in mind that β_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:

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

Regularization: Simplest setting

The least squares estimator:

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

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

Regularization: Simplest setting

The least squares estimator:

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

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

Now *penalize* β from getting too large:

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

Regularization: Simplest setting

The least squares estimator:

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

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

Now *penalize* β from getting too large:

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

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

Regularization: Simplest setting

The least squares estimator:

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

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

Now *penalize* β from getting too large:

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

Solution: $\widehat{\beta} = \frac{y}{1+\lambda}$. As λ gets large, $\widehat{\beta}$ 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 on 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 $(\widehat{\theta}_n, \widehat{\pi}_1)$, define classification rule

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

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

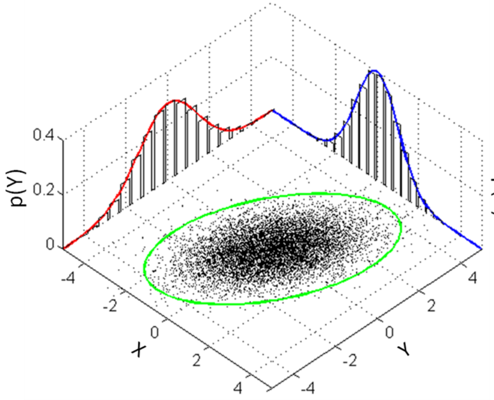
In the binary (two-class) case, we have two Gaussians:

$$
\begin{aligned}
X \mid y = 1 &\sim N(\mu_1, \Sigma_1) \\
X \mid y = 0 &\sim N(\mu_0, \Sigma_0)
\end{aligned}
$$

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`

Linear discriminant analysis

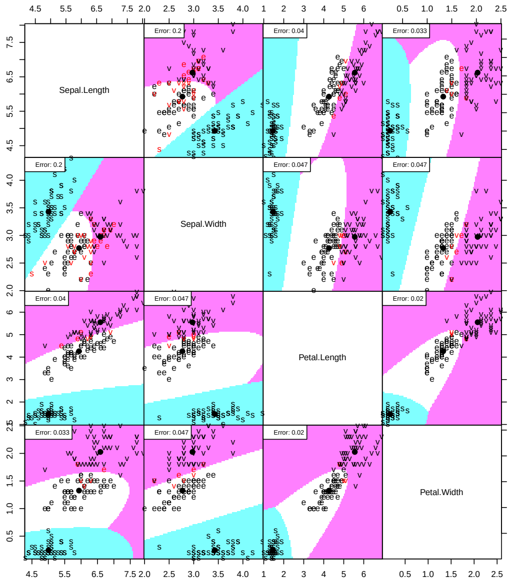
In the binary (two-class) case, we again have two Gaussians:

$$X \mid y = 1 \sim N(\mu_1, \Sigma)$$
$$X \mid y = 0 \sim N(\mu_0, \Sigma)$$

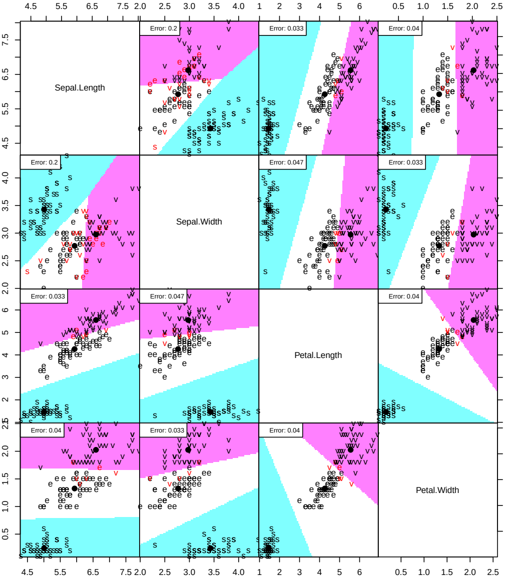
But now we use the same covariance matrix for each.

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.

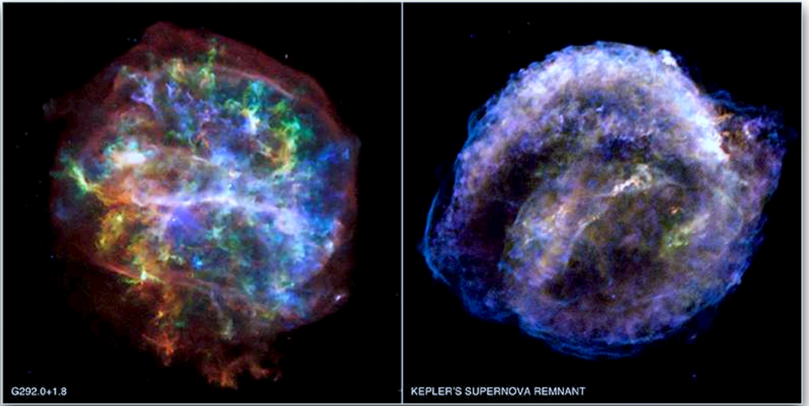
We only model the conditional probability $p(Y \mid X)$.

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

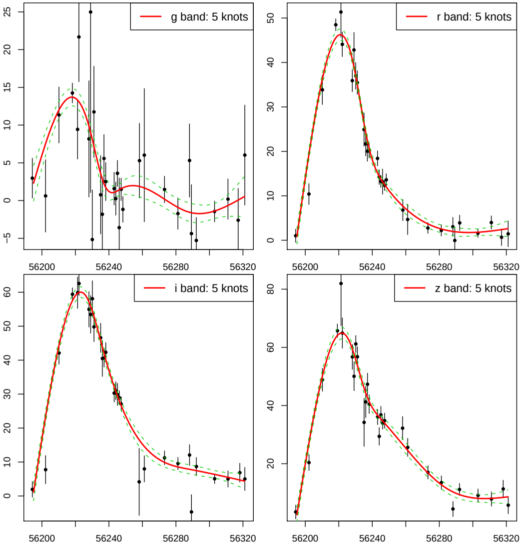


Two supernova remnants from the NASA's Chandra X-ray Observatory study. The right one is Type Ia. (Credit: NASA/CXC/UCSC/L. Lopez et al.)

Supernovae

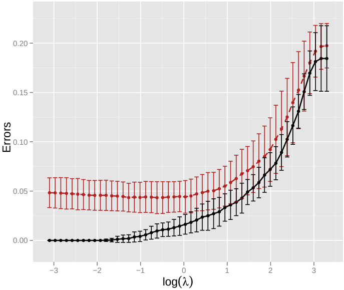
- Data are 20,000 real and simulated supernovae.
- For each supernova, there are a few noisy measurements of the flux (brightness) in four different filters — g-band (green), r-band (red), i-band (infrared) and z-band (blue).
- These noisy data are processed to fit a curve through the measurements in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

Supernovae

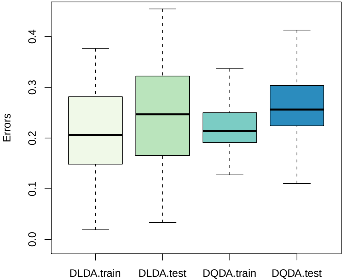


g band: 5 knots
r band: 5 knots
i band: 5 knots
z band: 5 knots
56200
56240
56280
56320

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 & discriminitive
- 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 β^2 that keeps them from being too big. *Shrinks* coefficients toward zero.