S&DS 265 / 565

Introductory Machine Learning

Classification (continued)

September 17

Yale

Upcoming items

- Assn 1 due Thursday (midnight, 11:59pm)
- Submit *both* your `.ipynb` notebook and a printout as `.pdf` (save as HTML then print as pdf).
- Quiz 2 will be posted next Thursday
- Questions?

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

Recall: Important concepts

Classification risk, or *error rate* is the probability of making a prediction mistake for a new data point.

The *empirical classification error* or *training error* is fraction of errors on the training data.

Recall: Important concepts

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

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

(where $\mathbb{1}$ just means 1 if the argument is true, 0 otherwise)

Optimal classification rule

The most accurate rule possible is called the *Bayes rule*.

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

The Bayes decision rule

From Bayes' theorem

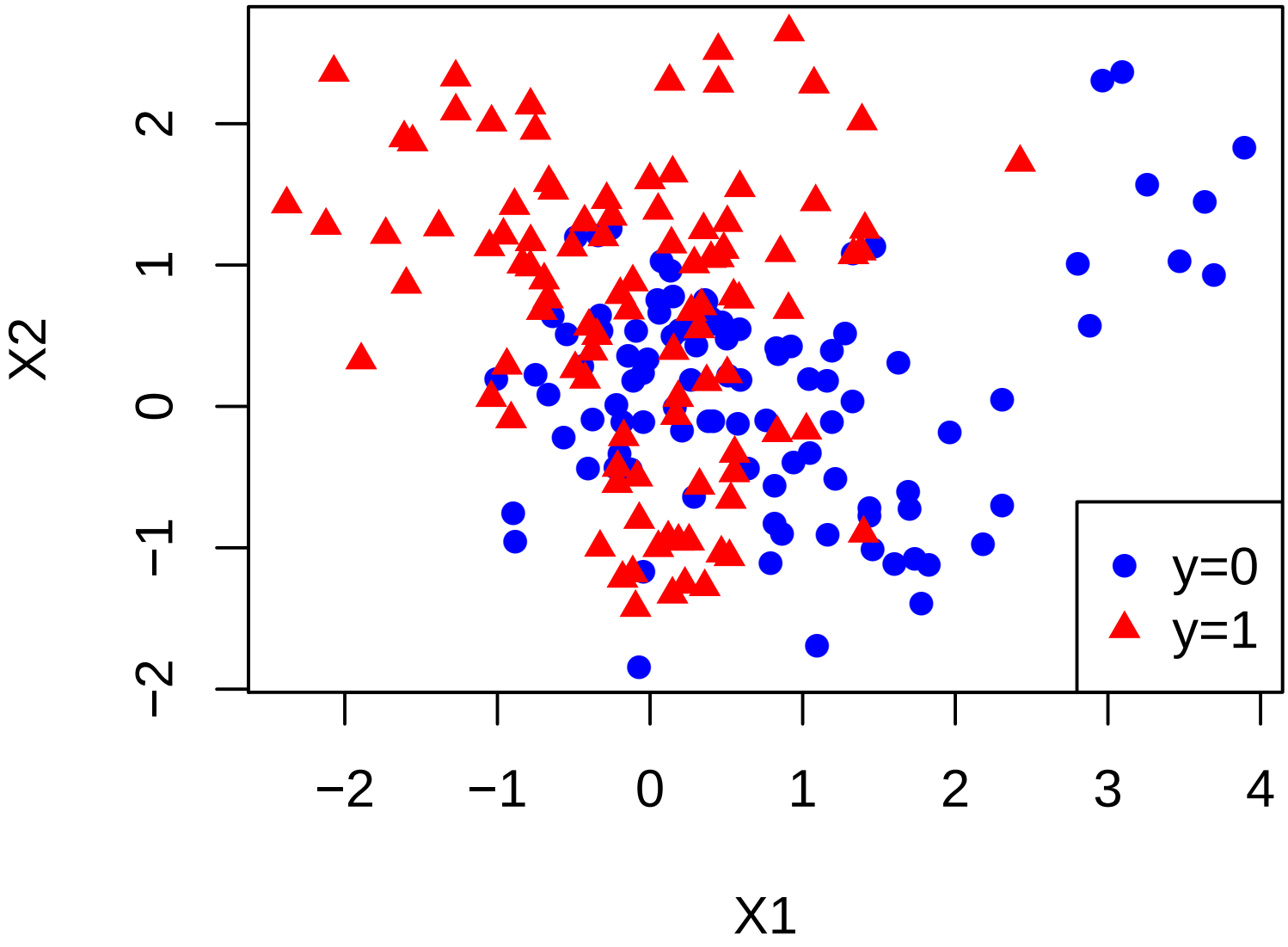
$$\mathbb{P}(Y = 1 \mid X = x) \quad = \quad \frac{p(x \mid Y = 1)\mathbb{P}(Y = 1)}{p(x \mid Y = 1)\mathbb{P}(Y = 1) + p(x \mid Y = 0)\mathbb{P}(Y = 0)}$$

So,

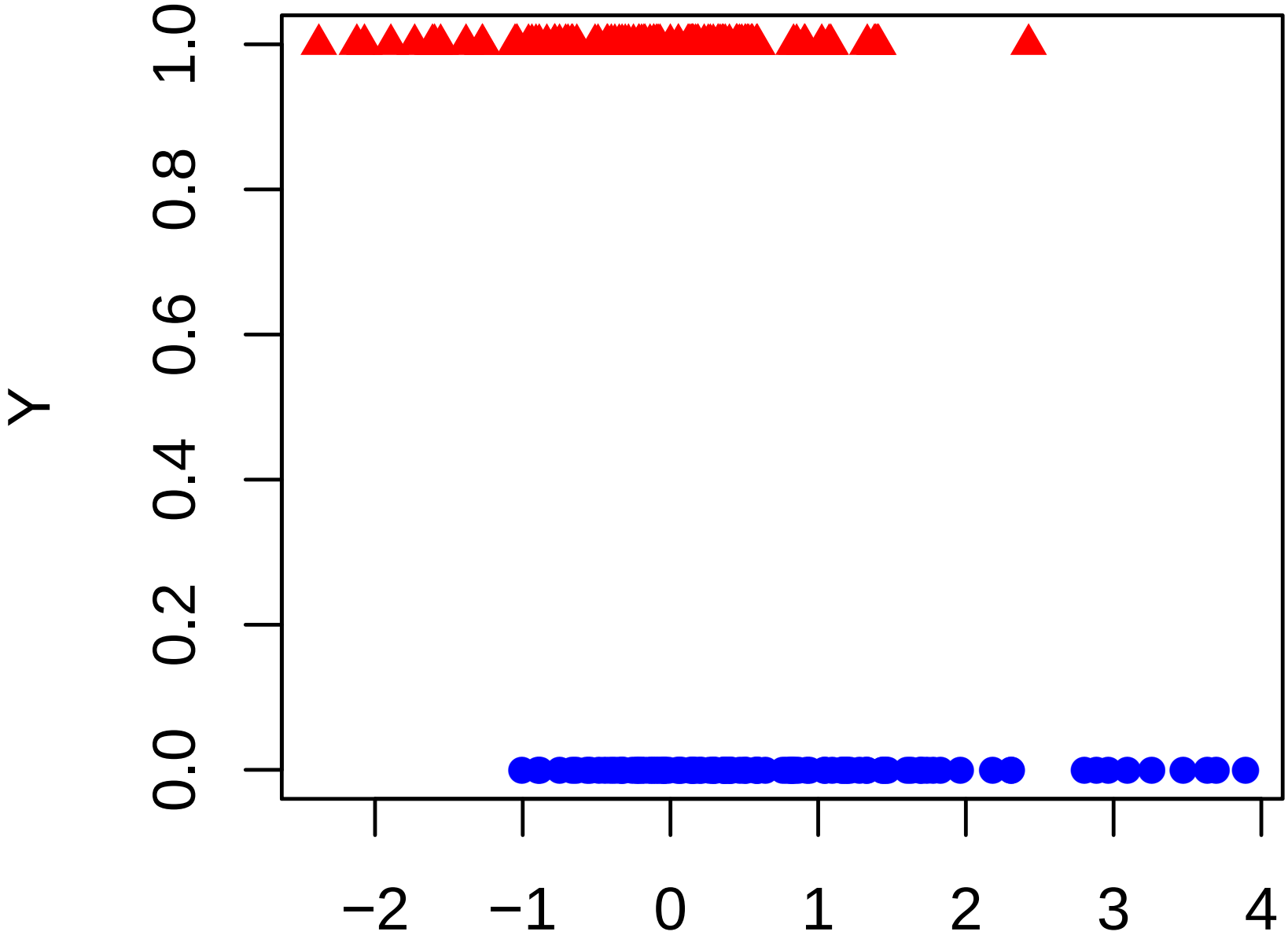
$$h^*(x) = 1 \quad \text{is equivalent to} \quad \frac{p(x \mid Y = 1)}{p(x \mid Y = 0)} > \frac{\mathbb{P}(Y = 0)}{\mathbb{P}(Y = 1)}.$$

These are population probabilities. Conceptually, the Bayes rule has access to all possible data.

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

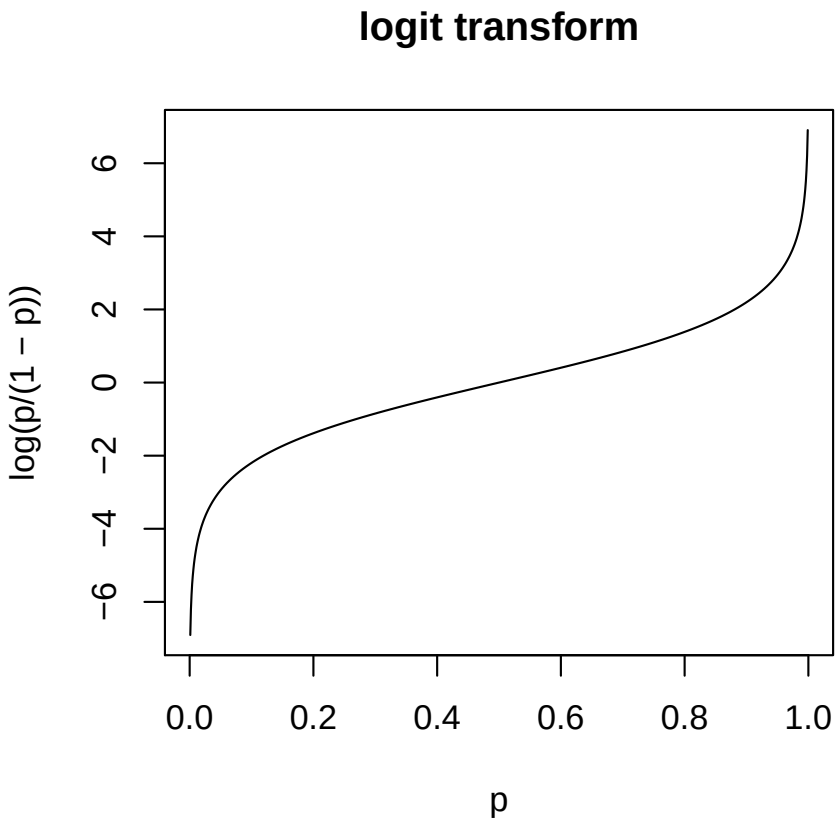
Conditional probabilities of the class:

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

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

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

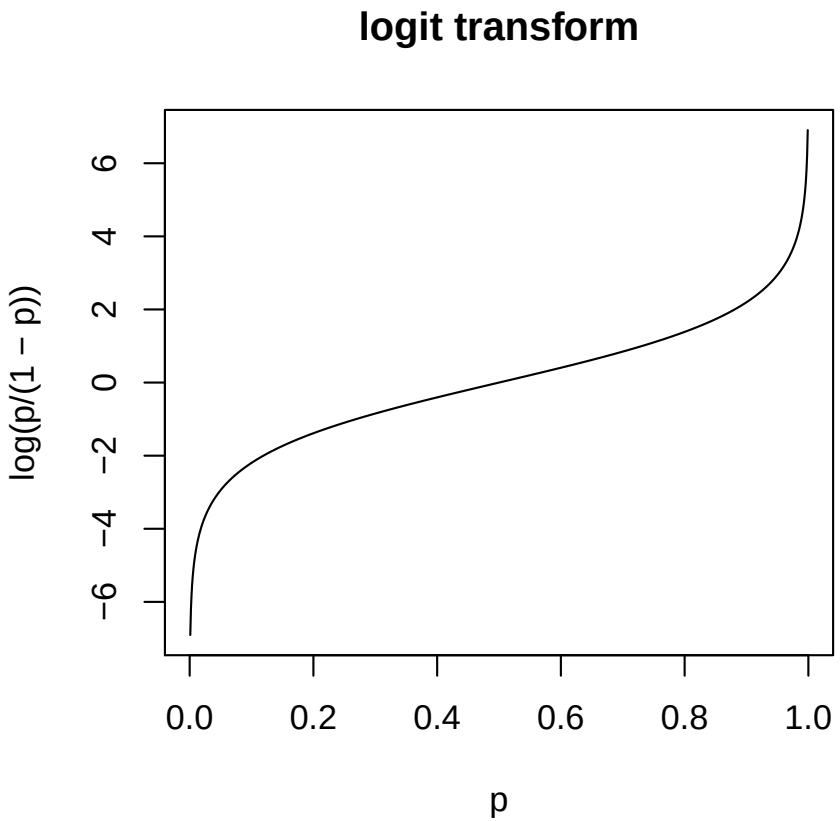
Logistic regression



The *logit* transform:

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

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:

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

Demo

Is this mushroom edible? Try it!

Have you every gone for a walk in the woods and been tempted to bring back some mushrooms for your dinner salad? Well, you may then wonder how many edible or poisonous mushrooms you might need to be shown before you could confidently forage in the forest on your own. We'll first load a database of mushrooms that have been hand-classified according to whether or not they are poisonous. Then, we'll put it into a form suitable for logistic regression by converting all of the categorical variables to "dummy" variables using a "1-hot" representation. Then we'll fit logistic regression models on training sets of increasing size.

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
```

Extension to more than 2 classes

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

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

Extension to more than 2 classes

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

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

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

Loss function for 3 classes

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

$$-\sum_{i=1}^{n} \log P(Y = y_i \mid X = x_i)$$

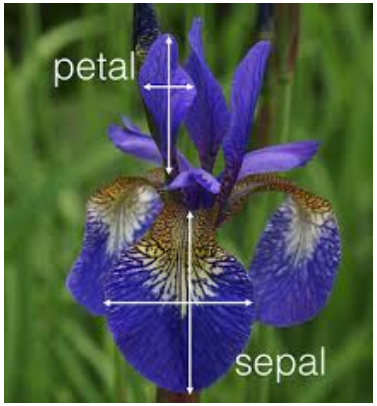
$$= \sum_{i=1}^{n} \left\{ \log(1 + e^{\beta_1^T x_i} + e^{\beta_2^T x_i}) - \beta_{y_i}^T x_i \right\}$$

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).



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.

Regularization

More generally, we can *penalize* a model with a penalty

$$\lambda\|\beta\|^2 = \lambda(\beta_1^2 + \beta_2^2 + \cdots + \beta_p^2)$$

that prevents the coefficients from being too large.

For the Iris problem, this would be

$$\begin{aligned}\lambda\|\beta\|^2 = \lambda(&\beta^2_{\text{setosa; p-width}} + \beta^2_{\text{setosa; p-length}} \\ &+ \beta^2_{\text{setosa; s-width}} + \beta^2_{\text{setosa; s-length}} \\ &+ \beta^2_{\text{versacolor; p-width}} + \beta^2_{\text{versacolor; p-length}} \\ &+ \beta^2_{\text{versacolor; s-width}} + \beta^2_{\text{versacolor; s-length}})\end{aligned}$$

Demo

Flower power: Logistic regression on the Iris data



We'll now carry out logistic regression for classification of the iris data.

Two flavors of classifiers

Generative: model both the input X and the output Y.

Discriminative: model only the output Y given X.

Two flavors of classifiers

Generative: model both the input X and the output Y.

Discriminative: model only the output Y given X.

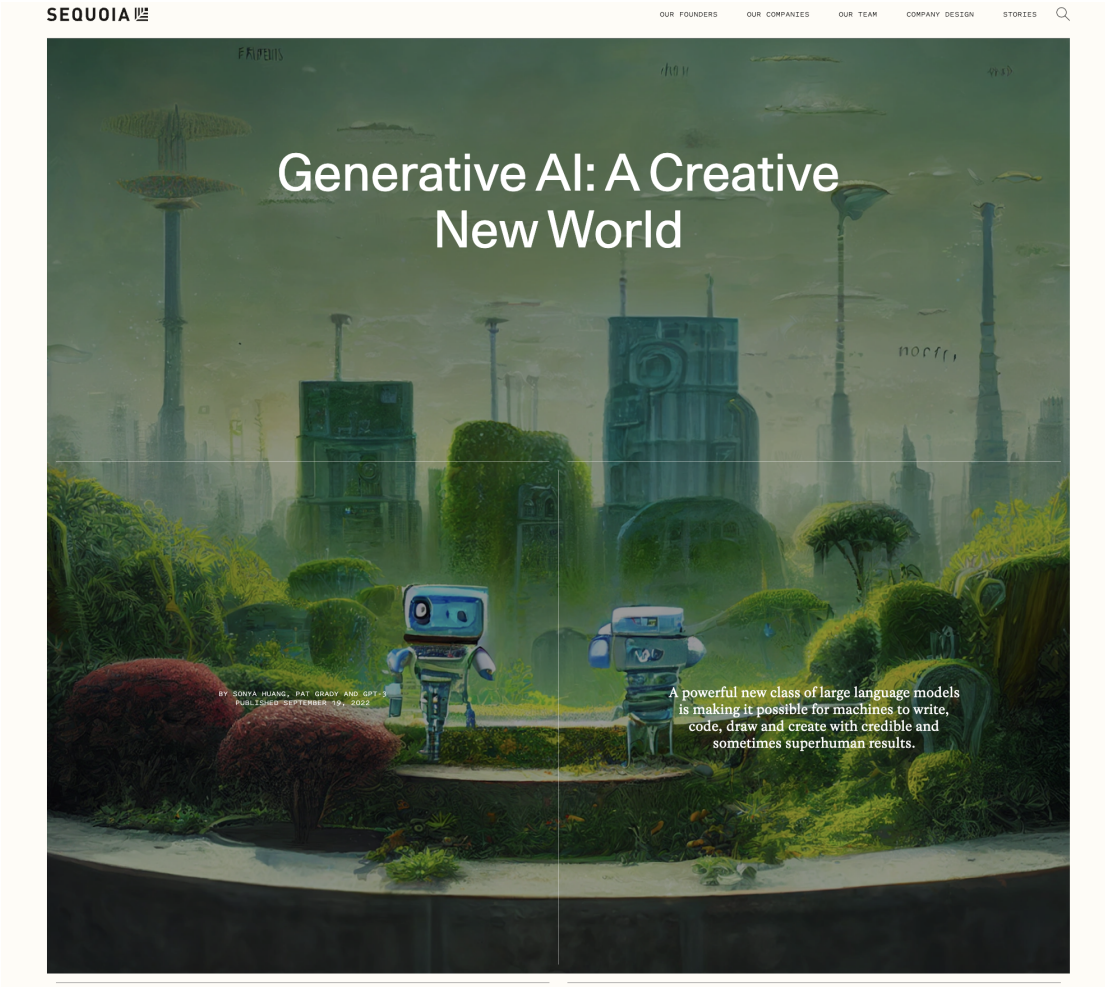
Which do you think is better?

Generative models

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

In the generative case we typically estimate the model by trying to maximize the *joint likelihood* $p(x, y)$ of the training data

Generative models



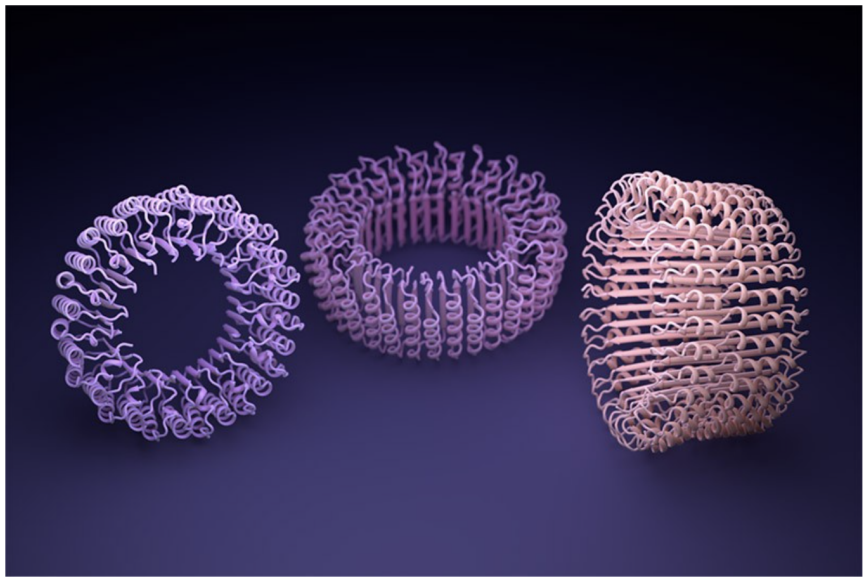
Generative models

NEWS | 15 September 2022

Scientists are using AI to dream up revolutionary new proteins

Huge advances in artificial intelligence mean researchers can design completely original molecules in seconds instead of months.

Ewen Callaway



Artificial-intelligence tools are helping to scientists to come up with proteins that are shaped unlike anything in nature. Credit: Ian C Haydon/UW Institute for Protein Design

Discriminative models

In the discriminative case we focus on the *conditional* distribution of the output given the input

We will typically estimate by maximizing the *conditional likelihood* of the training data

Bayes rule

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

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

But we can target the discriminant function directly

Gaussian discriminant analysis

- A type of generative model — *unfortunate terminology!*
- We model the inputs x using Gaussians
- Two flavors: Linear and Quadratic

Quadratic discriminant analysis

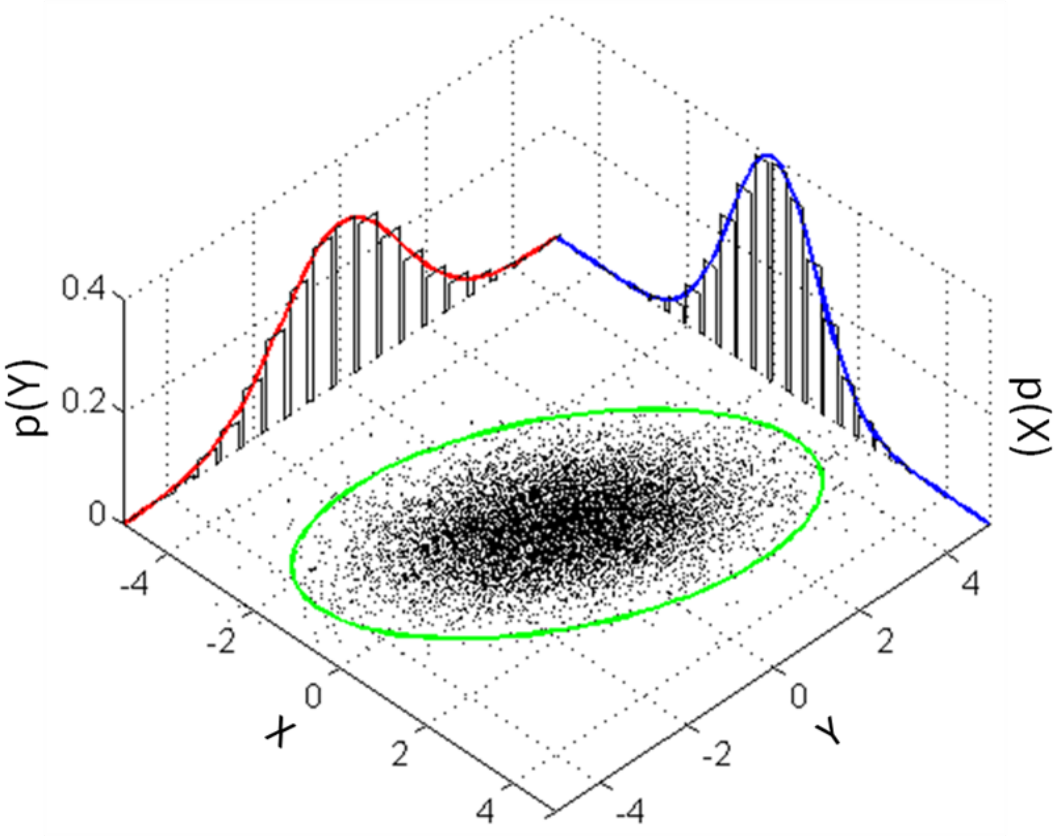
In the binary (two-class) case, we have two Gaussians:

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

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

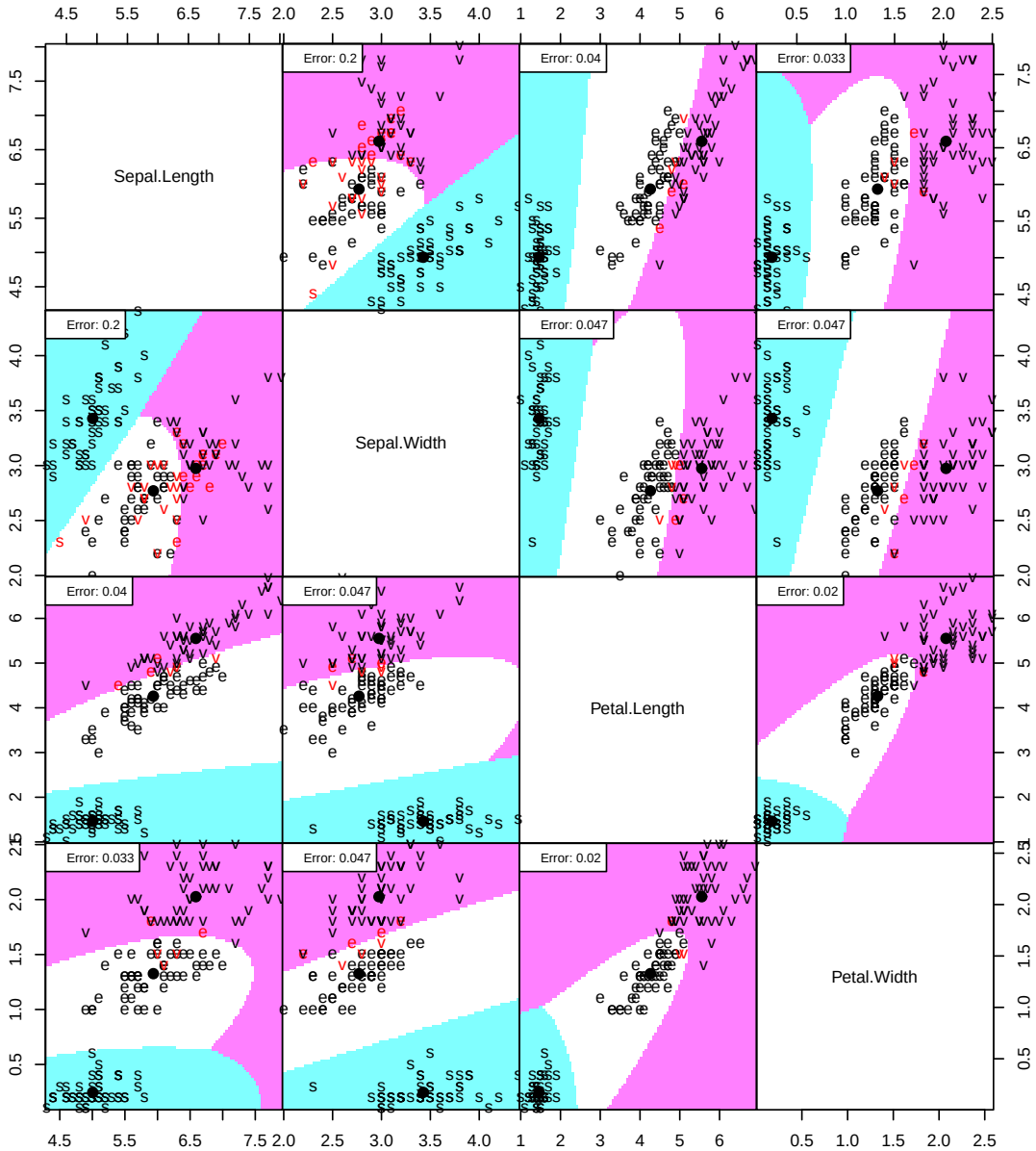
In the binary (two-class) case, we again have two Gaussians:

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

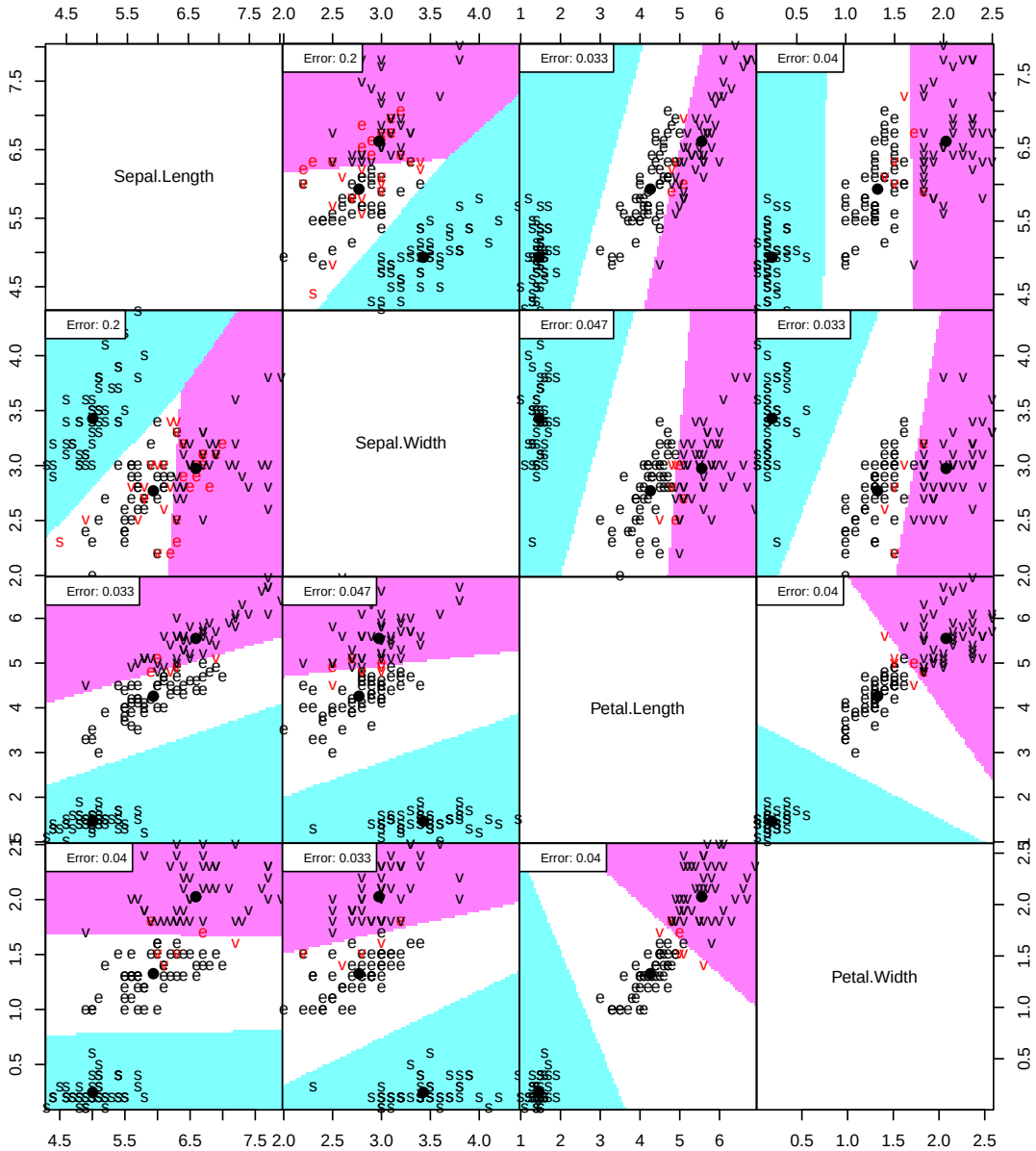
But now we use the *same covariance* matrix for each.

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.

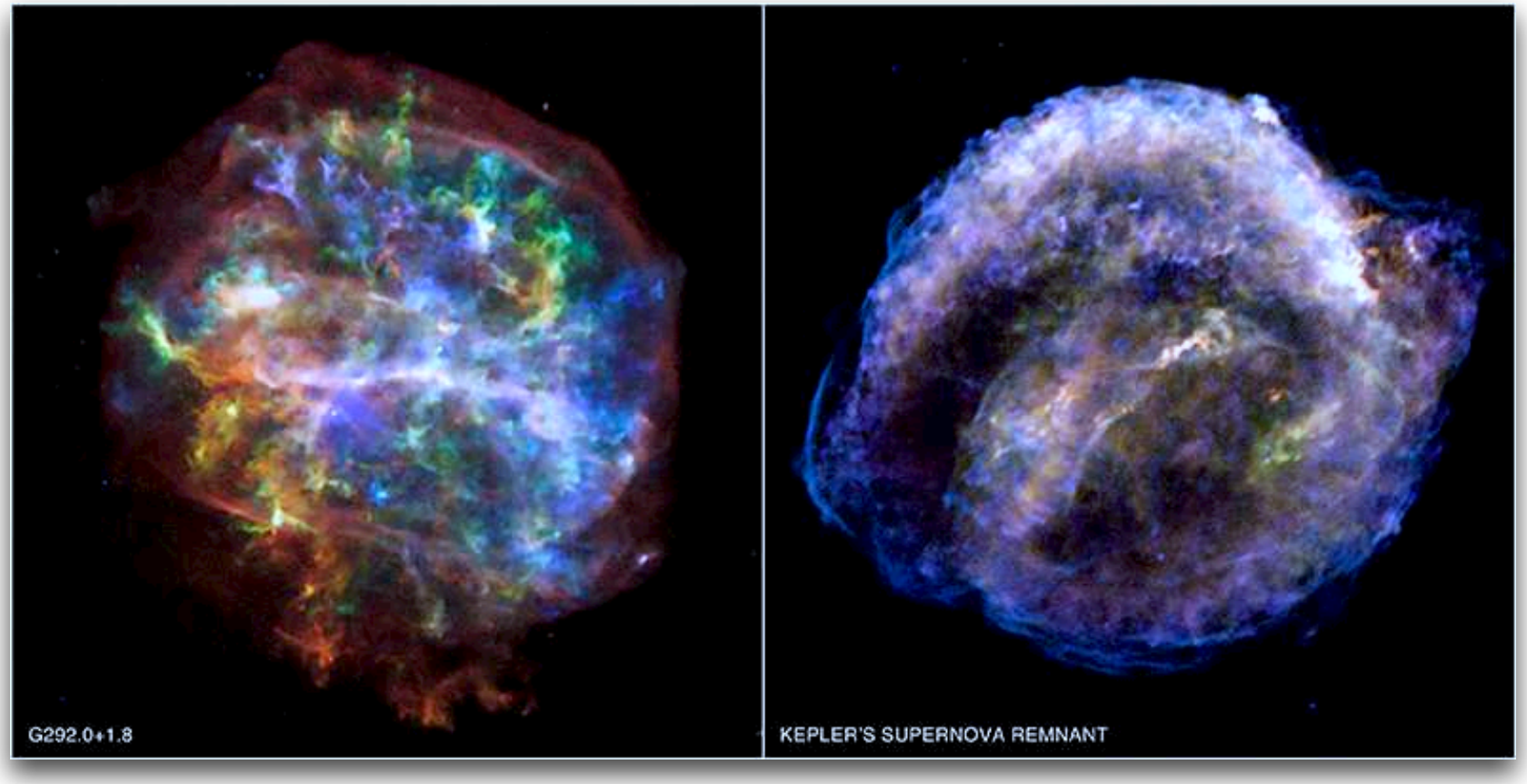
We only model the conditional probability $p(Y \mid X)$.

Logistic regression is the discrimintive version of linear discriminant analysis (the latter is a generative model–unfortunate terminology!)

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

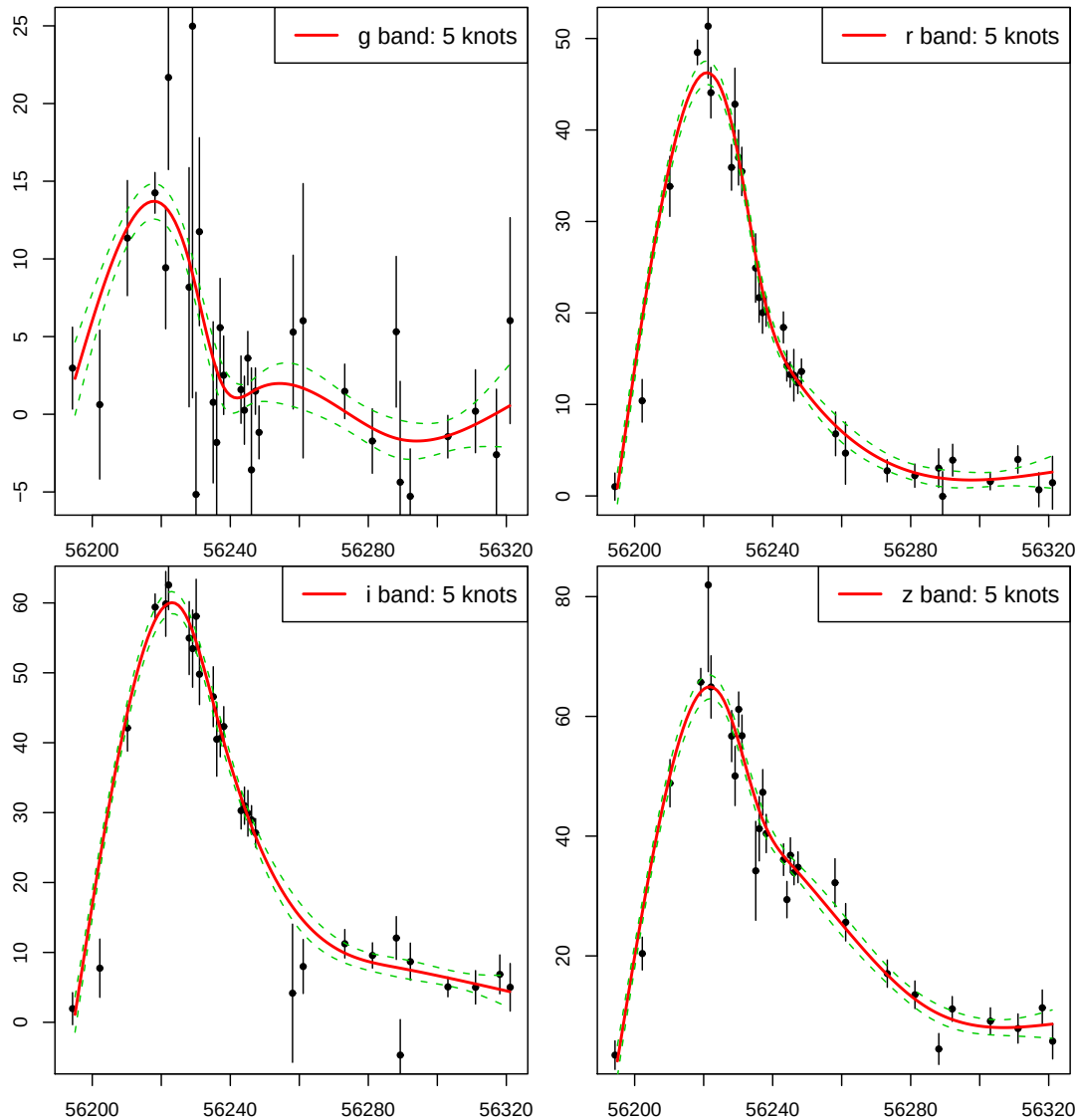


Two supernova remnants from the NASA's Chandra X-ray Observatory study. The right one is Type Ia. (Credit: NASA/CXC/UCSC/L. Lopez et al.)

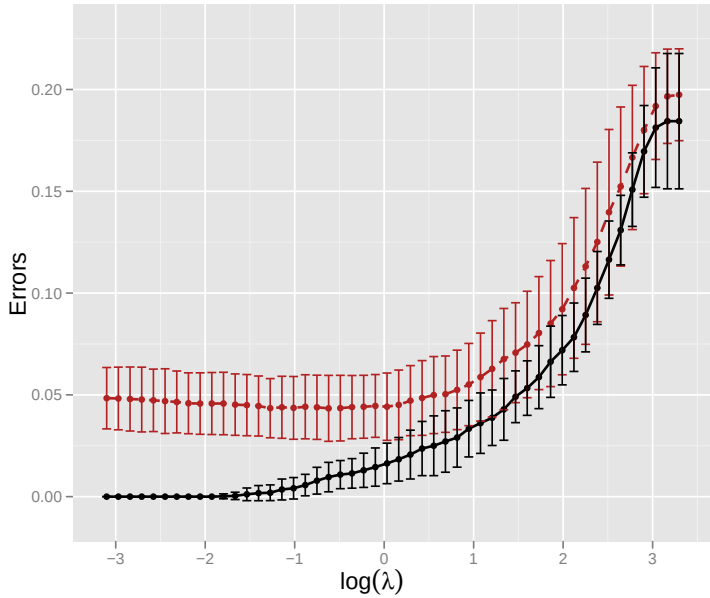
Supernovae

- 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, the values along this curve are used as predictor variables

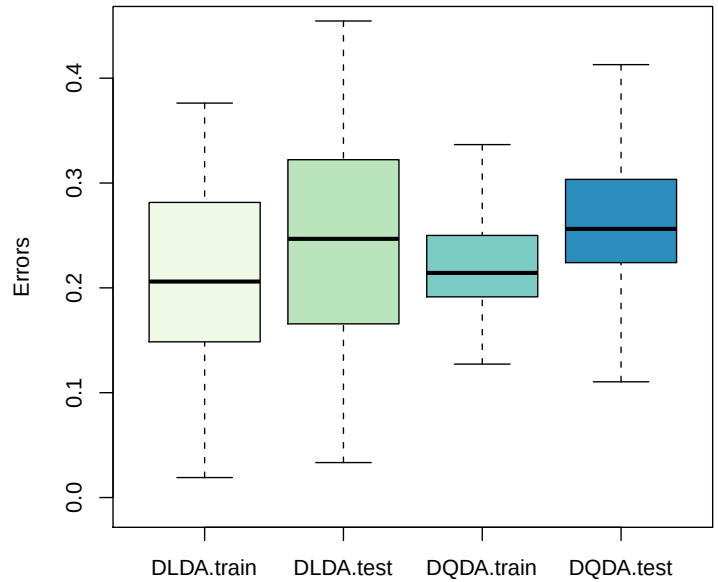
Supernovae


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

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.

Looking ahead

- Deep learning is used for classification
  - Discriminative models
  - Last layer is equivalent to logistic regression
  - Other layers extract predictive features
- Generative AI has led to a resurgence of generative models
  - Models can accurately model (generate) complex data like faces, human conversation, and natural images in the world