

Linear Classification Models

Stott Parker

UCLA

April 25, 2014

2D Gaussians

A **p -dimensional Gaussian function** has the form:

$$g(\mathbf{x}, \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{p/2}} \frac{1}{\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right).$$

where \mathbf{x} be a p -dimensional value, $\boldsymbol{\mu}$ is a p -dimensional vector of means, and Σ is a positive definite $p \times p$ covariance matrix.

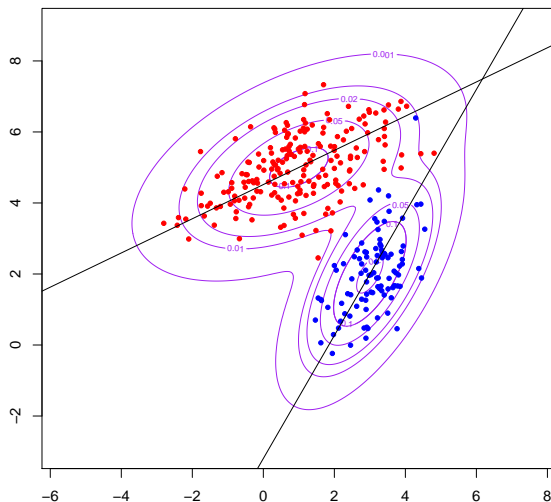
(We require Σ to be positive definite so the determinant is positive.)

Equivalently, if $W = \Sigma^{-1}$:

$$g(\mathbf{x}, \boldsymbol{\mu}, W^{-1}) = \frac{1}{(2\pi)^{p/2}} \sqrt{\det W} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})' W (\mathbf{x} - \boldsymbol{\mu})\right).$$

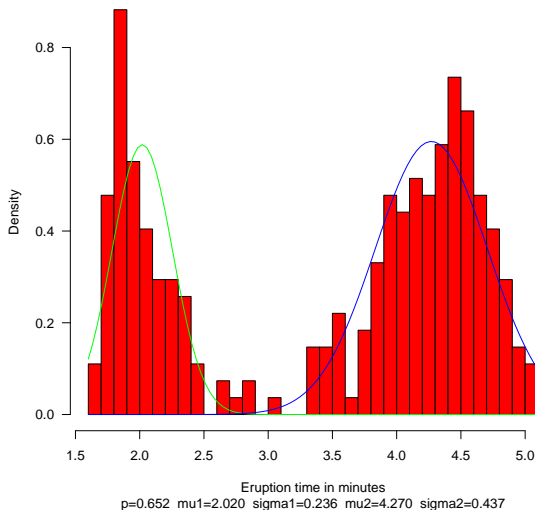
Multiple Gaussians

random 2D Gaussian Mixture with specified means and covariance



Old Faithful

Distribution of eruption times of Old Faithful



Mixture Models

To the distribution of geyser eruption times we can fit a *mixture model*

$$\text{Prob}[\text{eruption time} = x \mid \theta] = (1 - p) g(x, \mu_1, \sigma_1) + p g(x, \mu_2, \sigma_2).$$

This is a simple mixture of two gaussians, and so has five parameters:
 $p, \mu_1, \sigma_1, \mu_2, \sigma_2$.

If we can formalize the objective function, we can use an optimization method to find the optimal parameter vector

$$\theta = (p, \mu_1, \sigma_1, \mu_2, \sigma_2)$$

A commonly-used objective function: **maximum likelihood**.

Maximum Likelihood Estimation of Parameters

If the set of eruption times is $D = \{ t_i \mid i = 1, \dots, n \}$, then

$$\text{Likelihood}[\boldsymbol{\theta} \mid D] = \text{Prob}[D \mid \boldsymbol{\theta}] = \prod_{i=1}^n \text{Prob}[\text{eruption time} = t_i \mid \boldsymbol{\theta}]$$

and thus

$$\log \text{Likelihood}[\boldsymbol{\theta} \mid D] = \sum_{i=1}^n \log \left((1 - p) g(t_i, \mu_1, \sigma_1) + p g(t_i, \mu_2, \sigma_2) \right).$$

We give this objective function to an optimizer.

Classification Models

A **classification** of a feature vector \mathbf{x} is a value y in a set C of **classes**.

Given a training set of feature vectors \mathbf{x} and classes y ,
the **classification problem** is to find a function f such that $y = f(\mathbf{x})$.

Given a $n \times p$ matrix/dataset X (n observations of p -feature vectors), and
a $n \times 1$ vector \mathbf{y} of classifications, find a **classification model** $y = f(\mathbf{x})$
that minimizes the **loss function**

$$L(\mathbf{y}, f(X)) = \|\mathbf{y} - f(X)\|.$$

This restates the classification problem as an optimization problem.

Linear Discriminants

Using least squares we solve $X\beta = \mathbf{y}$ for a $p \times 1$ vector of coefficients β .

The classifier function can then be

$$y = f(\mathbf{x}) = \sigma(\langle \mathbf{x}, \beta \rangle)$$

where σ is a function that ‘rounds’ or ‘truncates’ to an integer class value.

If a constant intercept c is desired so $\beta = (\mathbf{w} \ c)$, then

$$\langle (\mathbf{x} \ 1), (\mathbf{w} \ c) \rangle = \langle \mathbf{w}, \mathbf{x} \rangle + c$$

— and the classifier is defined by the **hyperplane** $\langle \mathbf{w}, \mathbf{x} \rangle = c$.

In the two-class case, for example, we could use:

$$\sigma(t) = \text{sgn}(t) = \begin{cases} +1 & t > 0 \\ -1 & t < 0. \end{cases}$$

Input feature vectors \mathbf{x} will be given the classification $\sigma(\langle \mathbf{w}, \mathbf{x} \rangle + c)$.

Least Squares Linear Discriminants

We can find the coefficients \mathbf{w} and c via Least Squares as follows:

```
A = cbind( X, 1 )
# add a column of '1' values for an intercept coefficient

wc = solve(t(A) %*% A) %*% t(A) %*% y
# alternatively: wc = lsfit(X, y)$coefficients
# alternatively: wc = lm(y ~ X)$coefficients

w = wc[1:2]
c = wc[3]
classifier = function(x, w, c) { 2 * ((x %*% w + c) > 0) - 1 }
# i.e., ((x %*% w + c) > 0) ? +1 : -1
# where +1 = red and -1 = blue,
## classifier = function(x, w, c) { sign(x %*% w + c) }

# plot the discriminant
curve( -(w[1] * x + c)/w[2], col="green", add=TRUE )
```

LDA (Linear Discriminant Analysis)

Assume our k classes have a common covariance matrix Σ :

$$g_k(\mathbf{x}) = 1/(2\pi)^{p/2} \sqrt{\det(\Sigma^{-1})} \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)' \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right)$$

Given an input \mathbf{x} , we can estimate

$$\text{Prob[class} = k \mid \mathbf{x} \text{]} = \frac{g_k(\mathbf{x}) p_k}{\sum_{\ell} g_{\ell}(\mathbf{x}) p_{\ell}}$$

where p_{ℓ} is the (prior) probability of \mathbf{x} belonging to class i . Then:

$$\begin{aligned} \log \frac{\text{Prob[class} = k \mid \mathbf{x} \text{]}}{\text{Prob[class} = \ell \mid \mathbf{x} \text{]}} &= \log \frac{g_k(\mathbf{x}) p_k}{g_{\ell}(\mathbf{x}) p_{\ell}} \\ &= \log \frac{p_k}{p_{\ell}} - \frac{1}{2} (\boldsymbol{\mu}_k' \Sigma^{-1} \boldsymbol{\mu}_k + \boldsymbol{\mu}_{\ell}' \Sigma^{-1} \boldsymbol{\mu}_{\ell}) + (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{\ell})' \Sigma^{-1} \mathbf{x} \end{aligned}$$

which is a linear function of \mathbf{x} !

LDA (Linear Discriminant Analysis)

We can equivalently define the discriminant function

$$d_k = \log p_k - \frac{1}{2} \boldsymbol{\mu}_k' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \mathbf{x}' \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k$$

so that: $d_k > d_\ell \iff \text{Prob}[\text{class} = k | \mathbf{x}] > \text{Prob}[\text{class} = \ell | \mathbf{x}]$.

Thus we can define a classifier function

$$f(\mathbf{x}) = \operatorname{argmax}_k d_k(\mathbf{x}).$$

For a training set X , if we are not given values for p_k , $\boldsymbol{\mu}_k$, or $\boldsymbol{\Sigma}$:

- estimate p_k as the proportion of training examples in class k
- estimate $\boldsymbol{\mu}_k$ as the average of the \mathbf{x} examples in class k
- estimate $\boldsymbol{\Sigma}$ to be the covariance matrix of the training set X .

QDA (Quadratic Discriminant Analysis)

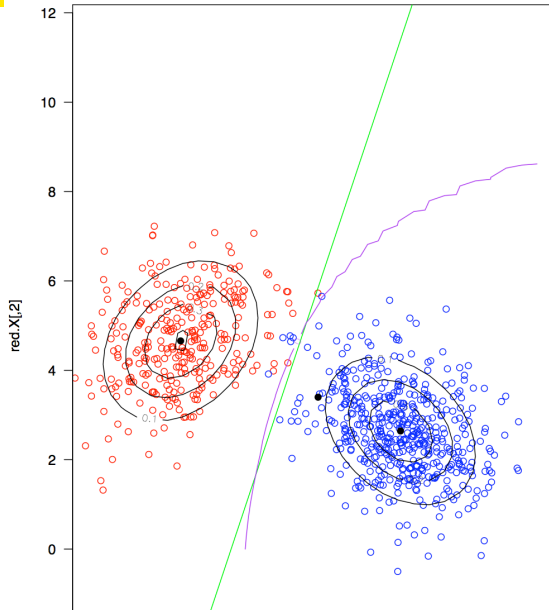
QDA generalizes on LDA by permitting discriminant functions of the form

$$d_k = \log p_k - \frac{1}{2} \log \det(\Sigma_k) - \frac{1}{2} \boldsymbol{\mu}_k' \Sigma_k^{-1} \boldsymbol{\mu}_k + (\mathbf{x} - \boldsymbol{\mu}_k)' \Sigma_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)$$

where each class k has its own covariance matrix Σ_k .

This is a quadratic function of \mathbf{x} .

QDA



Logistic Regression

Suppose there are two classes: $y = 0$ and $y = 1$.
We are given \mathbf{x} and seek to find the best value of y .

Instead of using a normal regression model like

$$y = \langle \mathbf{w}, \mathbf{x} \rangle + c$$

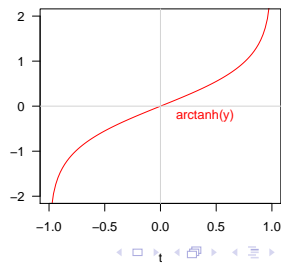
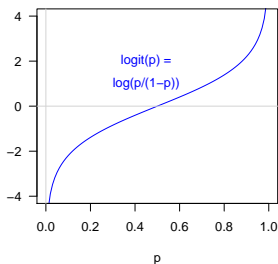
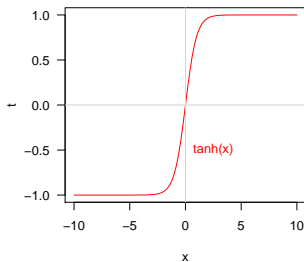
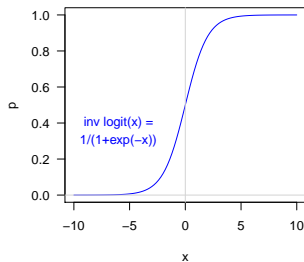
since y is discrete we want to use a model like

$$y = \sigma(\langle \mathbf{w}, \mathbf{x} \rangle + c)$$

where σ is a sigmoid function.

In **logistic regression**, σ is the inverse logistic function (logit^{-1}).

Sigmoid Functions



Sigmoid Functions

We say $\sigma(x)$ is a **sigmoid** function if it resembles the sign function $\text{sgn}(x)$. Important examples:

$$\begin{aligned}\sigma(x) &= \text{logit}^{-1}(cx) = \frac{1}{1 + e^{-cx}} \\ \sigma(x) &= \tanh(cx) = \frac{1 - e^{-2cx}}{1 + e^{-2cx}}\end{aligned}$$

Here c is a constant near 1 that can be used to 'tighten' the sigmoid.

These are both differentiable approximations of the sign function.

The logit function has an interesting property: it is the **log odds** function

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

Logistic Regression

Again, the target variable y is a variable with discrete values 0 and 1.

Since $\text{Prob}[\text{class} = 1 \mid \mathbf{x}] = 1 - \text{Prob}[\text{class} = 0 \mid \mathbf{x}]$, we can compute

$$\log \frac{\text{Prob}[\text{class} = 1 \mid \mathbf{x}]}{1 - \text{Prob}[\text{class} = 1 \mid \mathbf{x}]}$$

and choose class 1 if it is positive, otherwise choose class 0.

In other words, using the logit function

$$\sigma^{-1}(p) = \text{logit}(p) = \log \frac{p}{1-p}$$

we want to pick class

$$\text{round}(\text{logit}(\text{Prob}[\text{class} = 1 \mid \mathbf{x}])) \approx \sigma^{-1}(\text{Prob}[\text{class} = 1 \mid \mathbf{x}]).$$

Logistic Regression

In general, if there are $K > 2$ classes, we consider all K models of the form

$$\log \frac{\text{Prob}[\text{class} = k \mid \mathbf{x}]}{1 - \text{Prob}[\text{class} = k \mid \mathbf{x}]} = \langle \mathbf{w}_k, \mathbf{x} \rangle + c_k$$

with $\ell \neq k$ and σ is the logistic function.

An equivalent formulation of these models is:

$$\text{Prob}[\text{class} = k \mid \mathbf{x}] = \sigma(\langle \mathbf{w}_k, \mathbf{x} \rangle + c_k).$$

Logistic Regression

