Gaussian Discriminant Analysis

Note: Unless otherwise noted all references including images are from the required textbook, Machine Learning: A Probabilistic Perspective by Kevin P. Murphy.

Multivariate Gaussian

The **multivariate Gaussian** or **multivariate normal (MVN)** is the most widely used joint probability density function for continuous variables. The pdf of the MVN in D dimensions is defined by the following:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

Multivariate Gaussian

Where

$\boldsymbol{\mu} = \mathbb{E}[x] \in \mathbb{R}^D$ is the mean vector,

$\boldsymbol{\Sigma} = \mathrm{cov}[\mathrm{x}]$ is the $D \times D$ covariance matrix, and

the expression inside the exponent is the Mahalanobis distance between a data vector **x** and the mean vector $\boldsymbol{\mu}$

MLE for an MVN

If there are N independent and identically distributed samples $\mathbf{x}_i \sim \mathcal{N}(\mu, \Sigma)$, then the MLE for the parameters is given by

$$\hat{\boldsymbol{\mu}}_{mle} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \triangleq \bar{\mathbf{x}}$$

$$\hat{\boldsymbol{\Sigma}}_{mle} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T = \frac{1}{N}(\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T) - \bar{\mathbf{x}}\,\bar{\mathbf{x}}^T$$

That is, the MLE is just the empirical mean and empirical covariance.

MLE for an MVN

In the univariate case, the MLE is

$$\hat{\mu} = \frac{1}{N}\sum_i x_i = \overline{x}$$

$$\hat{\sigma}^2 = \frac{1}{N}\sum_i (x_i - \overline{x})^2 = \left(\frac{1}{N}\sum_i x_i^2\right) - (\overline{x})^2$$

Gaussian Discriminant Analysis

One important application of MVNs is to define the class conditional densities in a generative classifier

$$p(\mathbf{x}|y=c,\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

The resulting technique is called (Gaussian) **discriminant analysis** or **GDA**.

Gaussian Discriminant Analysis

We can classify a feature vector using the following decision rule

$$\hat{y}(\mathbf{x}) = \underset{c}{\operatorname{argmax}} \left[\log p(y = c|\boldsymbol{\pi}) + \log p(\mathbf{x}|\boldsymbol{\theta}_c) \right]$$

Gaussian Discriminant Analysis

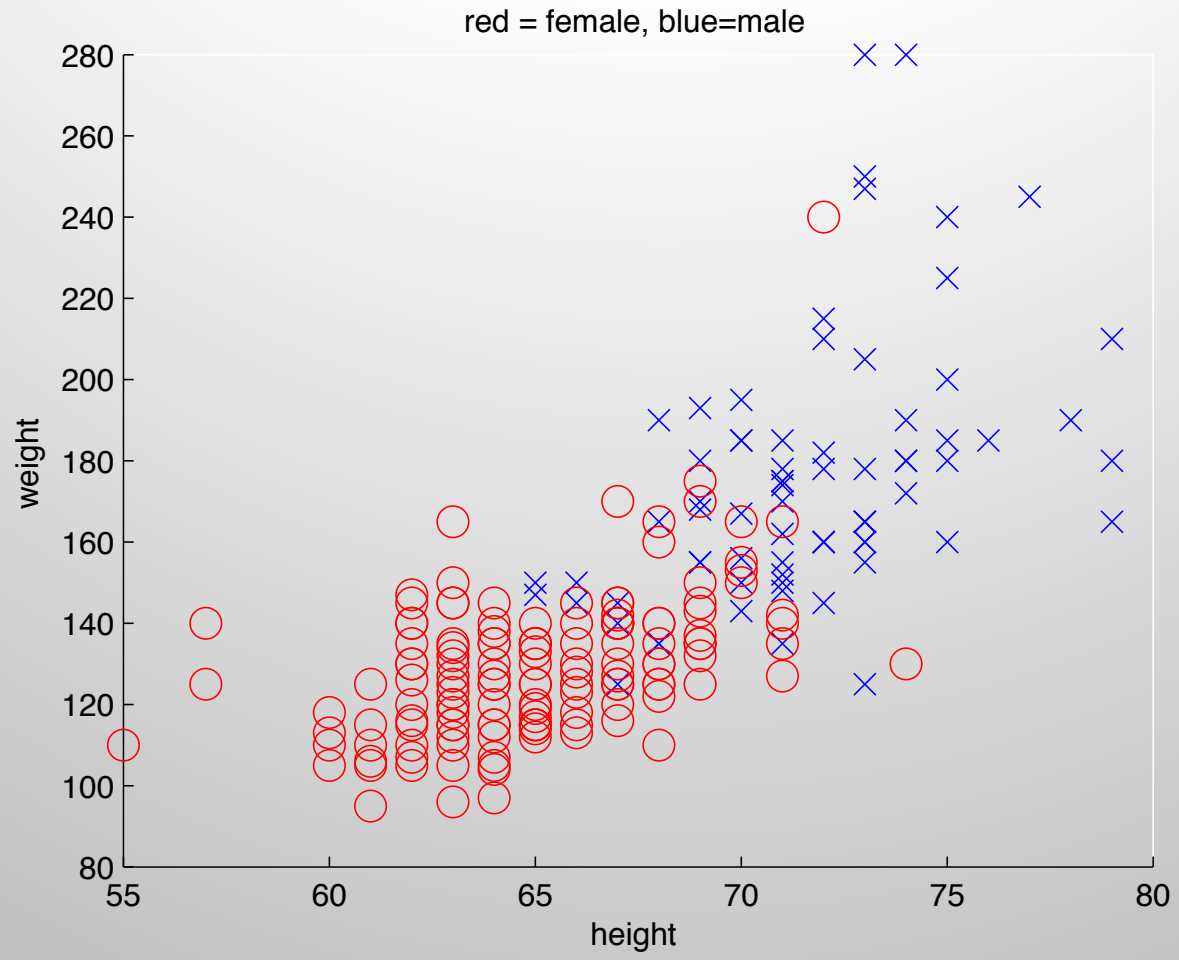
When we compute the probability of **x** under each class conditional density, we are measuring the distance from **x** to the center of each class, μ_c, using the Mahalanobis distance. This can be thought of as a **nearest centroids classifier**.

Gaussian Discriminant Analysis

If there is a uniform prior over classes, we can classify a new test vector as

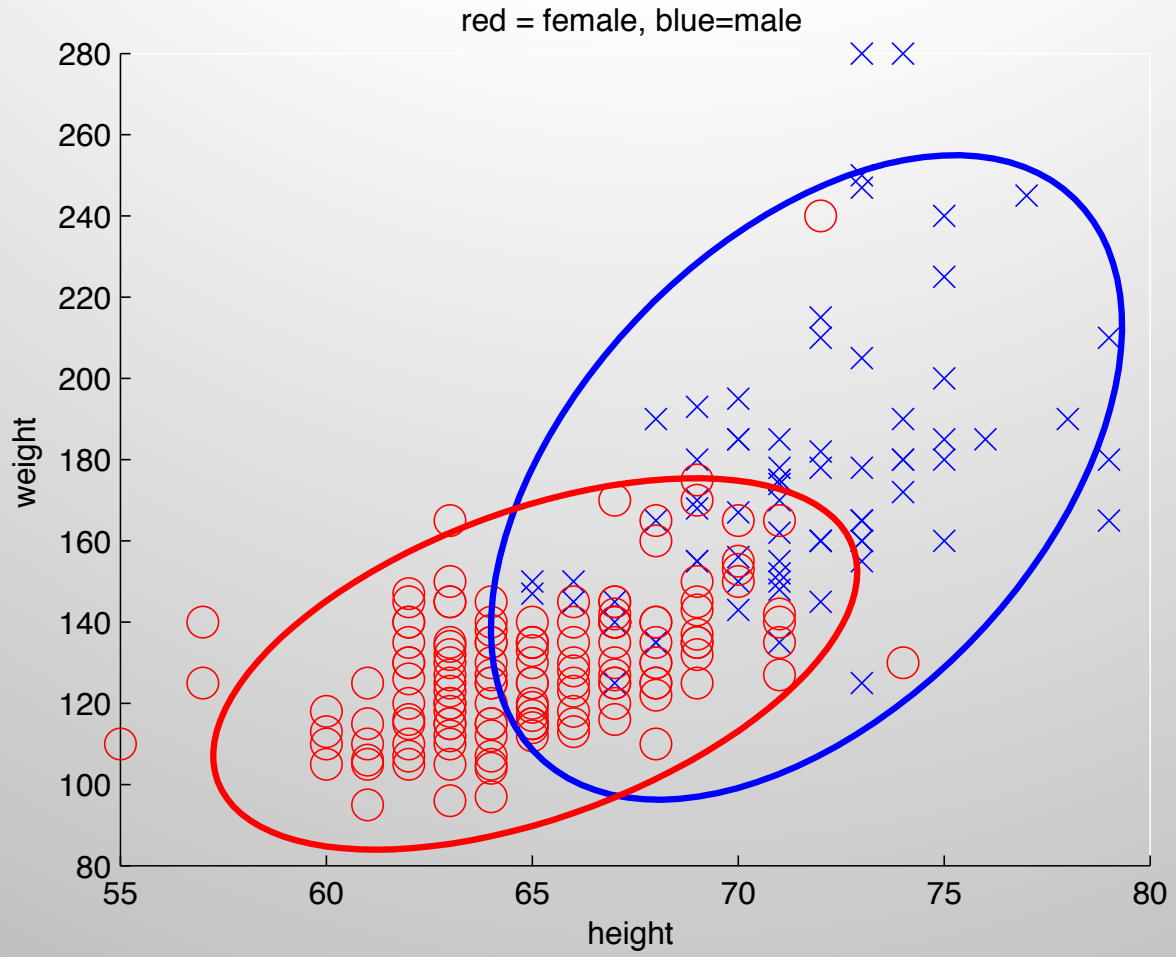
$$\hat{y}(\mathbf{x}) = \underset{c}{\operatorname{argmin}} (\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (\mathbf{x} - \boldsymbol{\mu}_c)$$

Gaussian Discriminant Analysis



Height and weight data of men and women.

Gaussian Discriminant Analysis



Visualization of 2d Gaussian fit to each class. 95% of the probability mass is inside the ellipse.

Quadratic Discriminant Analysis

In a generative classifier, the posterior over class labels is given by

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{p(y = c|\boldsymbol{\theta})p(\mathbf{x}|y = c, \boldsymbol{\theta})}{\sum_{c'} p(y = c'|\boldsymbol{\theta})p(\mathbf{x}|y = c', \boldsymbol{\theta})}$$

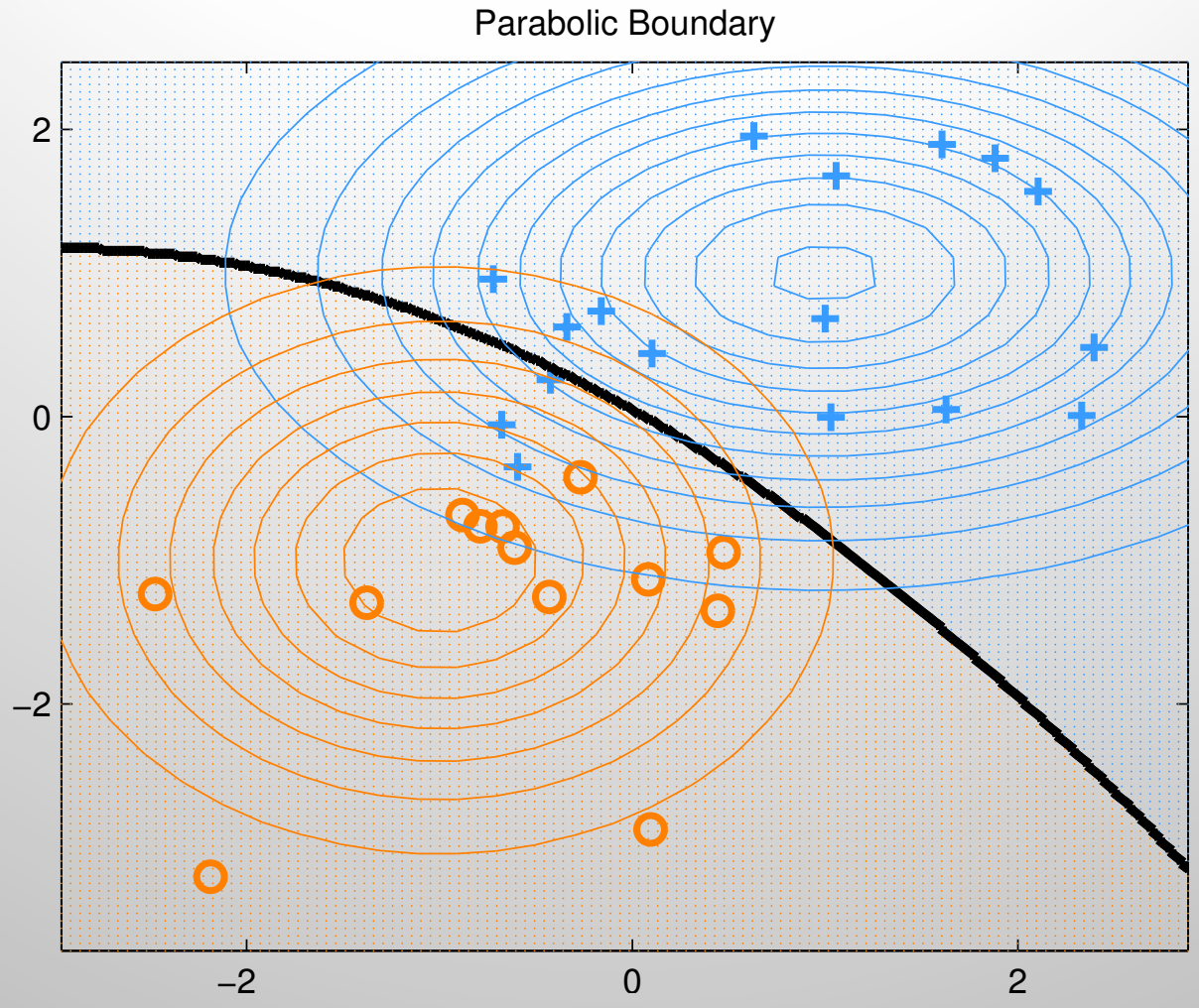
Quadratic Discriminant Analysis

Plugging in the Gaussian density gives

$$p(y=c|\mathbf{x},\boldsymbol{\theta}) = \frac{\pi_c|2\pi\boldsymbol{\Sigma}_c|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c)\right]}{\sum_{c'}\pi_{c'}|2\pi\boldsymbol{\Sigma}_{c'}|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{c'})^T\boldsymbol{\Sigma}_{c'}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{c'})\right]}$$

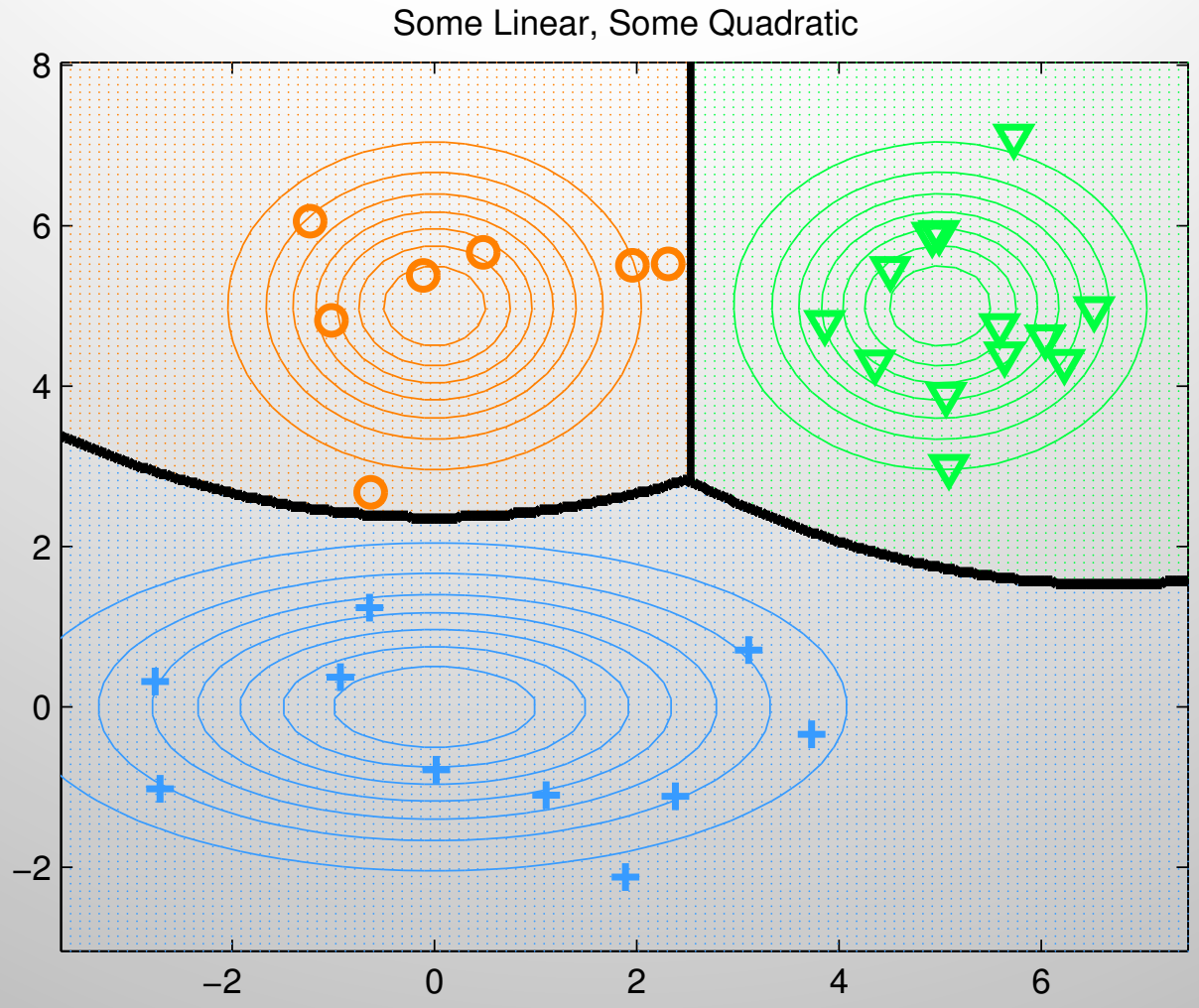
Thresholding this results in a quadratic function of **x**. The results is known as **quadratic discriminant analysis** or **QDA**.

Quadratic Discriminant Analysis



Quadratic decision boundary in 2d for 2-class case

Quadratic Discriminant Analysis



Quadratic decision boundaries in 2d for 3-class case

Linear Discriminant Analysis

Now consider a special case in which the covariance matrices are **tied** or **shared** across classes, $\Sigma_c = \Sigma$. In this case, the posterior is

$$
\begin{aligned}
p(y=c|\mathbf{x},\boldsymbol{\theta}) &\propto \pi_c \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\right] \\
&= \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c\right]\exp[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}]
\end{aligned}
$$

Since the quadratic term $\mathbf{x}^T\Sigma^{-1}\mathbf{x}$ is independent of c, it will cancel out in numerator and denominator.

Linear Discriminant Analysis

If we define

$$\begin{aligned} \gamma_c &= -\frac{1}{2}\boldsymbol{\mu}_c^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_c + \log \pi_c \\ \boldsymbol{\beta}_c &= \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_c \end{aligned}$$

then we can write

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_c^T \mathbf{x} + \gamma_c}}{\sum_{c'} e^{\boldsymbol{\beta}_{c'}^T \mathbf{x} + \gamma_{c'}}} = \mathcal{S}(\boldsymbol{\eta})_c$$

Linear Discriminant Analysis

Where

$\eta = [\beta_1^T \mathbf{x} + \gamma_1, \ldots, \beta_C^T \mathbf{x} + \gamma_C]$ and

$\mathcal{S}$ is the **softmax** function, defined as follows

$$\mathcal{S}(\eta)_c = \frac{e^{\eta_c}}{\sum_{c'=1}^{C} e^{\eta_{c'}}}$$

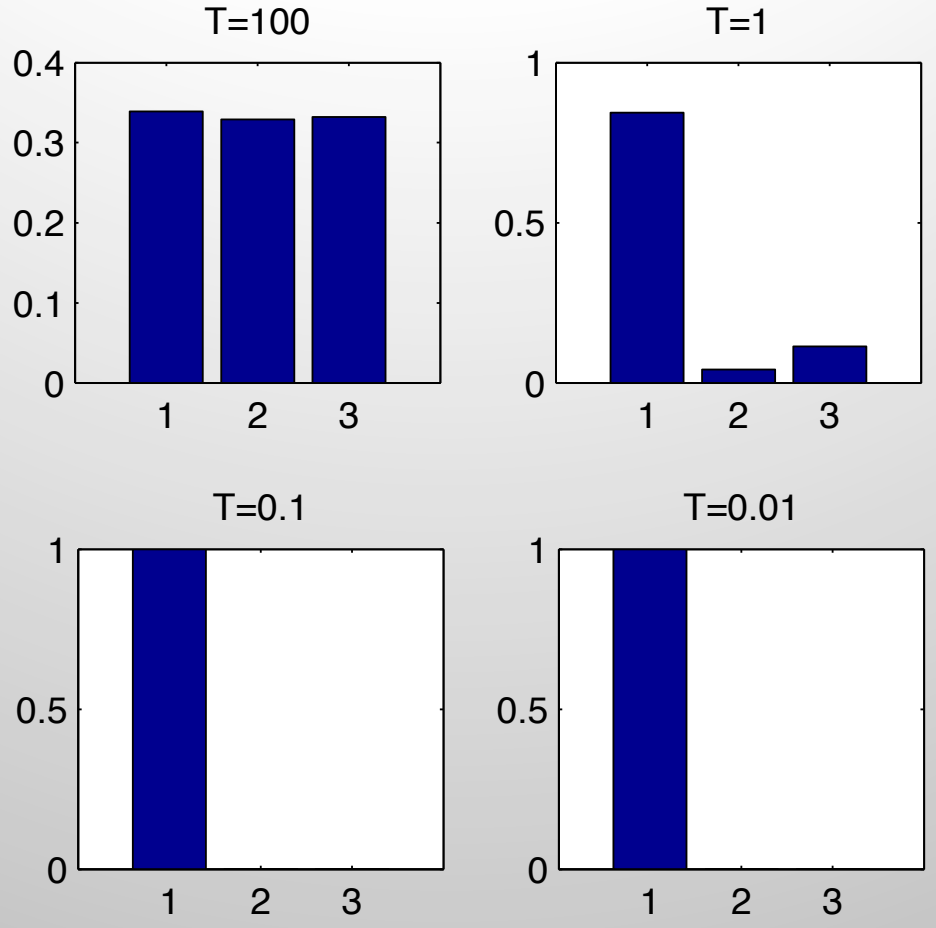
Linear Discriminant Analysis

The softmax function is so-called since it acts a bit like the max function. To see this, let us divide each η_c by a constant T called the temperature. Then as $T \to 0$, we find

$$\mathcal{S}(\boldsymbol{\eta}/T)_c = \begin{cases} 1.0 & \text{if } c = \text{argmax}_{c'} \, \eta_{c'} \\ 0.0 & \text{otherwise} \end{cases}$$

In other words, at low temperatures, the distribution spends essentially all of its time in the most probable state, whereas at high temperatures, it visits all states uniformly.

Linear Discriminant Analysis



Softmax distribution $\mathcal{S}(\eta, T)$, where $\eta = (3,0,1)$ at different temperatures T.

Linear Discriminant Analysis

If we take logs in equation

$$p(y = c|\mathbf{x}, \boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_c^T \mathbf{x} + \gamma_c}}{\sum_{c'} e^{\boldsymbol{\beta}_{c'}^T \mathbf{x} + \gamma_{c'}}} = \mathcal{S}(\boldsymbol{\eta})_c$$

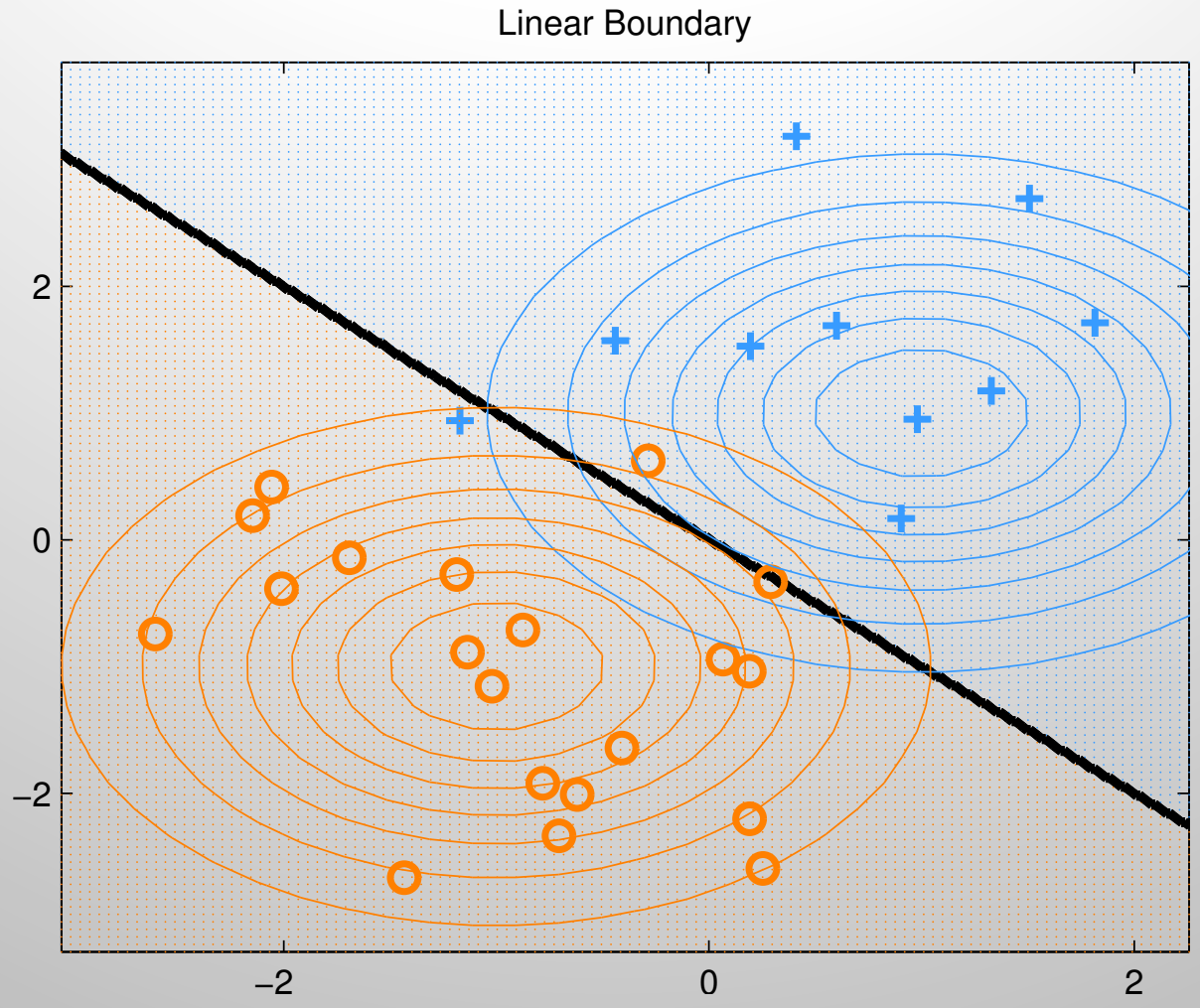
we end up with a linear function of $\mathbf{x}$. The reason it is linear is because the $\mathbf{x}^T \Sigma^{-1} \mathbf{x}$ cancels from the numerator and denominator. Thus the decision boundary between any two classes, say c and c', will be a straight line. Hence this technique is called **linear discriminant analysis** or **LDA**.

Linear Discriminant Analysis

We can derive the form of this line as follows

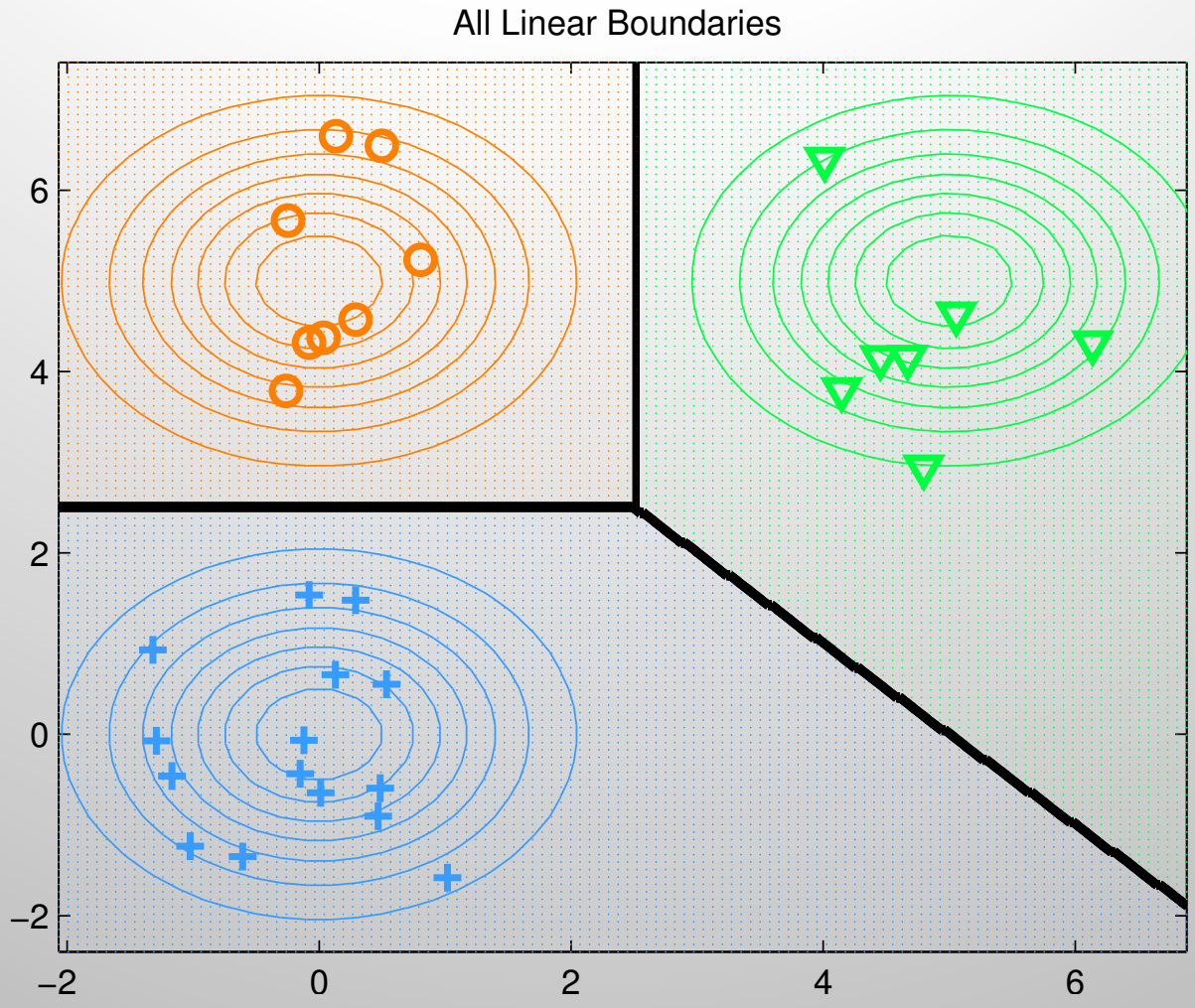
$$
\begin{aligned}
p(y = c|\mathbf{x}, \boldsymbol{\theta}) &= p(y = c'|\mathbf{x}, \boldsymbol{\theta}) \\
\boldsymbol{\beta}_c^T \mathbf{x} + \gamma_c &= \boldsymbol{\beta}_{c'}^T \mathbf{x} + \gamma_{c'} \\
\mathbf{x}^T (\boldsymbol{\beta}_{c'} - \boldsymbol{\beta}) &= \gamma_{c'} - \gamma_c
\end{aligned}
$$

Linear Discriminant Analysis



Linear decision boundary in 2d for 2-class case

Linear Discriminant Analysis



Linear decision boundary in 2d for 3-class case

MLE for Discriminant Analysis

How do we fit a discriminant analysis model? The simplest way is to use maximum likelihood. The log-likelihood function is as follows:

$$\log p(\mathcal{D}|\boldsymbol{\theta}) = \left[\sum_{i=1}^{N}\sum_{c=1}^{C}\mathbb{I}(y_i = c)\log \pi_c\right] + \sum_{c=1}^{C}\left[\sum_{i:y_i=c}\log \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)\right]$$

MLE for Discriminant Analysis

For the class prior, we have $\hat{\pi}_c = \frac{N_c}{N}$ as with naive Bayes.

For the class-conditional densities, we just partition the data based on its class label, and compute the MLE for each Gaussian:

$$\hat{\boldsymbol{\mu}}_c = \frac{1}{N_c} \sum_{i:y_i=c} \mathbf{x}_i, \quad \hat{\boldsymbol{\Sigma}}_c = \frac{1}{N_c} \sum_{i:y_i=c} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_c)^T$$

Strategies for Preventing Overfitting

The speed and simplicity of the MLE method is one of the greatest appeals. However, the MLE can badly overfit in high dimensions. In particular the MLE for a full covariance matrix is singular if $N_c < D$. And even when $N_c > D$, the MLE can be ill-conditioned, meaning it is close to singular. There are several possible solutions to this problem.

Strategies for Preventing Overfitting

- Use a diagonal covariance matrix for each class, which assumes the features and conditionally independent; this is equivalent to using a naive Bayes classifier.
- Use a full covariance matrix, but force it to be the same for all classes, $\Sigma_c = \Sigma$. This is an example of **parameter tying** or **parameter sharing**, and is equivalent to LDA.