Computation

For QDA we need to calculate:

$$\delta_k(\mathbf{x}) = -\frac{1}{2} log(|\Sigma_k|) - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + log(\pi_k)$$

Lets first consider the case that

$$\Sigma_k = I, \forall k \ .$$

This is the case where each distribution is spherical, around the mean point.

Case 1

When $\Sigma_k = I$

We have:

$$\delta_k = -\frac{1}{2} log(|I|) - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^\top I(\mathbf{x} - \boldsymbol{\mu}_k) + log(\pi_k)$$

but $\log(|I|) = \log(1) = 0$

and $(\mathbf{x} - \boldsymbol{\mu}_k)^\top I(\mathbf{x} - \boldsymbol{\mu}_k) = (\mathbf{x} - \boldsymbol{\mu}_k)^\top(\mathbf{x} - \boldsymbol{\mu}_k)$ is the squared Euclidean distance between two points $\mathbf{x}$ and $\boldsymbol{\mu}_k$

Thus under this condition (i.e. $\Sigma = I$) , a new point can be classified by its distance from the center of a class, adjusted by some prior.

Further, for two-class problem with equal prior, the discriminating function would be the perpendicular bisector of the 2-class's means.

Case 2

When $\Sigma_k \neq I$

Using the Singular Value Decomposition (SVD) of Σ_k

we get

$$\Sigma_k = U_k S_k U_k^\top$$

Note: Σ_k is a symmetric matrix $\Sigma_k = \Sigma_k^T$, so we have $\Sigma_k = U_k S_k U_k^\top$.

$$(\mathbf{x}-\boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)$$

$$\begin{aligned}
&= (\mathbf{x}-\boldsymbol{\mu}_k)^\top U_k S_k^{-1} U_k^T (\mathbf{x}-\boldsymbol{\mu}_k) \\
&= (U_k^\top \mathbf{x} - U_k^\top \boldsymbol{\mu}_k)^\top S_k^{-1} (U_k^\top \mathbf{x} - U_k^\top \boldsymbol{\mu}_k) \\
&= (U_k^\top \mathbf{x} - U_k^\top \boldsymbol{\mu}_k)^\top S_k^{-\frac{1}{2}} S_k^{-\frac{1}{2}} (U_k^\top \mathbf{x} - U_k^\top \boldsymbol{\mu}_k) \\
&= (S_k^{-\frac{1}{2}} U_k^\top \mathbf{x} - S_k^{-\frac{1}{2}} U_k^\top \boldsymbol{\mu}_k)^\top I (S_k^{-\frac{1}{2}} U_k^\top \mathbf{x} - S_k^{-\frac{1}{2}} U_k^\top \boldsymbol{\mu}_k) \\
&= (S_k^{-\frac{1}{2}} U_k^\top \mathbf{x} - S_k^{-\frac{1}{2}} U_k^\top \boldsymbol{\mu}_k)^\top (S_k^{-\frac{1}{2}} U_k^\top \mathbf{x} - S_k^{-\frac{1}{2}} U_k^\top \boldsymbol{\mu}_k)
\end{aligned}$$

For δ_k , the second term becomes what is also known as the Mahalanobis distance:

Think of $S_k^{-\frac{1}{2}} U_k^\top$ as a linear transformation that takes points in class k and distributes them spherically around a point, like in Case 1.

Thus when we are given a new point, we can apply the modified δ_k values to calculate $h^*(x)$. After applying the singular value decomposition, Σ_k^{-1} is considered to be an identity matrix such that

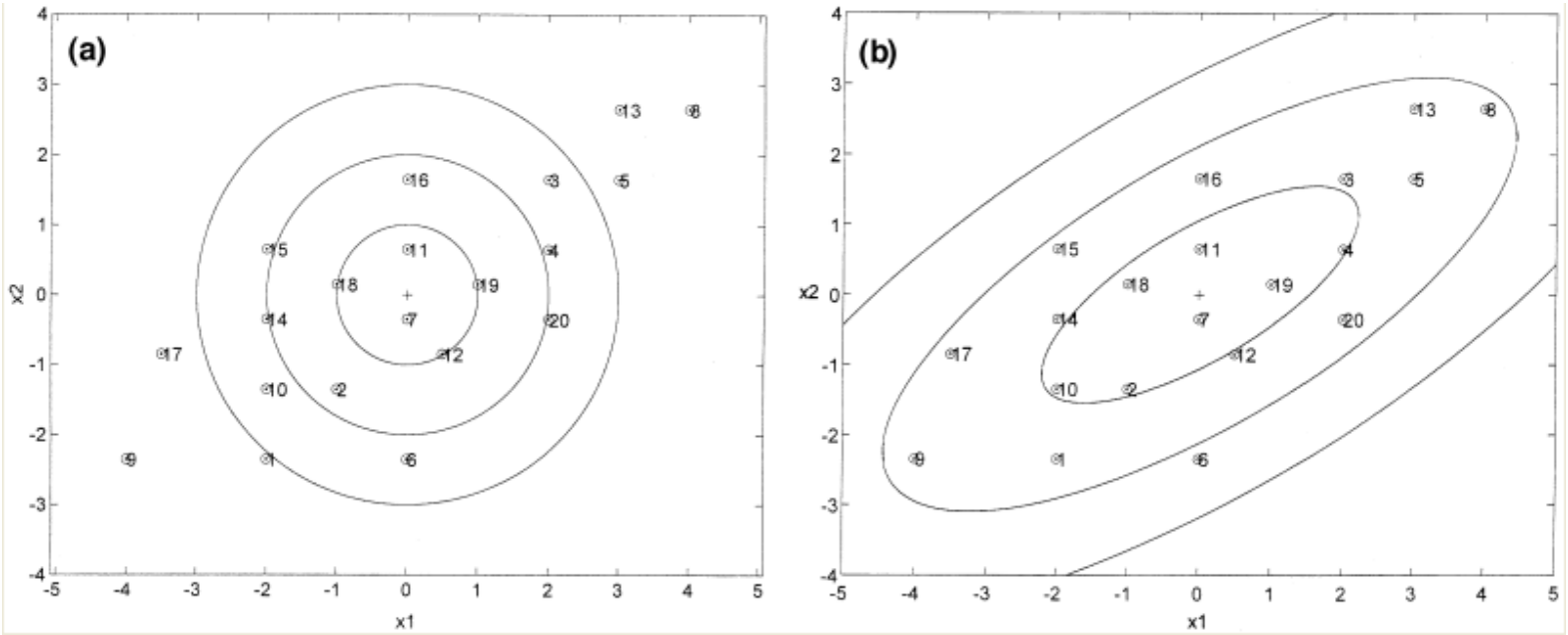
$$\delta_k = -\frac{1}{2} log(|I|) - \frac{1}{2}[(S_k^{-\frac{1}{2}} U_k^\top \mathbf{x} - S_k^{-\frac{1}{2}} U_k^\top \boldsymbol{\mu}_k)^\top (S_k^{-\frac{1}{2}} U_k^\top \mathbf{x} - S_k^{-\frac{1}{2}} U_k^\top \boldsymbol{\mu}_k)] + log(\pi_k)$$

and,

$$\log(|I|) = \log(1) = 0$$

For applying the above method with classes that have different covariance matrices (for example the covariance matrices Σ_0 and Σ_1 for the two class case), each of the covariance matrices has to be decomposed using SVD to find the according transformation. Then, each new data point has to be transformed using each transformation to compare its distance to the mean of each class (for example for the two class case, the new data point would have to be transformed by the class 1 transformation and then compared to μ_0 and the new data point would also have to be transformed by the class 2 transformation and then compared to μ_1).

The difference between Case 1 and Case 2 (i.e. the difference between using the Euclidean and Mahalanobis distance) can be seen in the illustration below.



Alternative approach to do QDA

There is a trick that allows us to use the linear discriminant analysis (LDA) algorithm to generate a quadratic function that can be used to classify data. This trick is similar to, but more primitive than, the Kernel trick that will be discussed later in the course.

In this approach the feature vector is augmented with the quadratic terms (i.e. new dimensions are introduced) We then apply LDA on the new higher-dimensional data.

The motivation behind this approach is to take advantage of the fact that fewer parameters have to be calculated in LDA , and therefore have a more robust classifier when we have fewer data points.

Using this trick, we introduce two new vectors, $\hat{\mathbf{w}}$ and $\hat{\mathbf{x}}$ such that:

$$\hat{\mathbf{w}} = [w_1, w_2, ..., w_d, v_1, v_2, ..., v_d]^T$$

and

$$\hat{\mathbf{x}} = [x_1, x_2, ..., x_d, {x_1}^2, {x_2}^2, ..., {x_d}^2]^T$$

We can then apply LDA to estimate the new function:
$\hat{g}(\mathbf{x}, \mathbf{x}^2) = \hat{y} = \hat{\mathbf{w}}^T \hat{\mathbf{x}}$.

Note that we can do this for any x and in any dimension; we could extend a $d \times n$ matrix to a quadratic dimension by appending another $d \times n$ matrix with the original matrix squared, to a cubic dimension with the original matrix cubed, or even with a different function altogether, such as a $sin(x)$ dimension. Note, we are not applying QDA, but instead extending LDA to calculate a non-linear boundary, that will be different from QDA.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a method of dimensionality reduction/feature extraction that transforms the data from a d-dimensional space into a new coordinate system of dimension p, where $p \leq d$ (the worst case would be to have $p = d$).

Principal Component Analysis (PCA)

- The goal is to preserve as much of the variance in the original data as possible in the new coordinate systems.
- Give data on d variables, the hope is that the data points will lie mainly in a linear subspace of dimension lower than d.
- In practice, the data will usually not lie precisely in some lower dimensional subspace.
- The new variables that form a new coordinate system are called **principal components** (PCs).

Principal Component Analysis (PCA)

- PCs are denoted by $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_d$.
- The principal components form a basis for the data.
- Since PCs are orthogonal linear transformations of the original variables there is at most d PCs.
- Normally, not all of the d PCs are used but rather a subset of p PCs, $\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_p$
- In order to approximate the space spanned by the original data points $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$ We can choose p based on what percentage of the variance of the original data we would like to maintain.

Principal Component Analysis (PCA)

The first PC, $\mathbf{u}_1$ is called **first principal component** and has the maximum variance, thus it accounts for the most significant variance in the data.

The second PC, $\mathbf{u}_2$ is called **second principal component** and has the second highest variance and so on until PC $\mathbf{u}_d$ which has the minimum variance.

Principal Component Analysis (PCA)

The most common definition of PCA, due to Hotelling is that, for a given set of data vectors x_i, $i \in 1...n$, the p principal axes are those orthonormal axes onto which the variance retained under projection is maximal.

Principal Component Analysis (PCA)

- In order to capture as much of the variability as possible, let us choose the first principal component, denoted by $\boldsymbol{u_1}$, to capture the maximum variance.
- Suppose that all centred observations are stacked into the columns of a $d \times n$ matrix X, where each column corresponds to a d-dimensional observation and there are n observations.
- The projection of n, d-dimensional observations on the first principal component $\boldsymbol{u_1}$ is $\boldsymbol{u_1}^T X$.

Principal Component Analysis (PCA)

We want projection on this first dimension to have maximum variance.

$$var(\boldsymbol{u_1}^T X) = \boldsymbol{u_1}^T S \boldsymbol{u_1}$$

where S is the $d \times d$ sample covariance matrix of X.

- Clearly $var(\boldsymbol{u_1}^T X)$ can be made arbitrarily large by increasing the magnitude of $\boldsymbol{u_1}$.
- $var(\boldsymbol{u_1}^T X) = \boldsymbol{u_1}^T S \boldsymbol{u_1}$ where S is sample covariance matrix of sample data X.
- This means that the variance stated above has no upper limit and so we can not find the maximum.
- To solve this problem, we choose $\boldsymbol{u_1}$ to maximize $\boldsymbol{u_1}^T S \boldsymbol{u_1}$ while constraining $\boldsymbol{u_1}$ to have unit length.
- Therefore, we can rewrite the above optimization problem as:

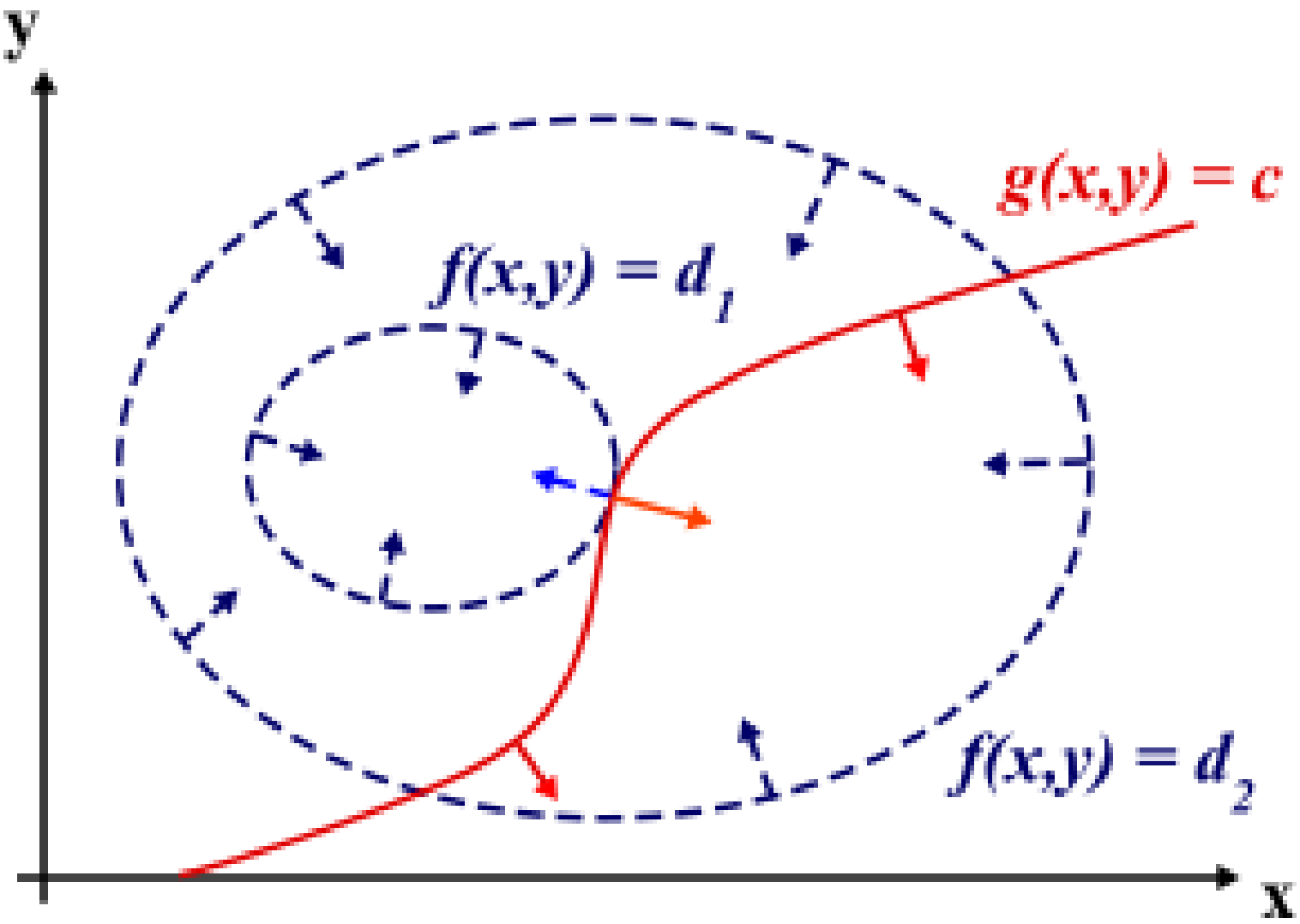
$$\max \ \boldsymbol{u_1}^T S \boldsymbol{u_1}$$
$$subject\ to\ \boldsymbol{u_1}^T \boldsymbol{u_1} = 1$$

To solve this optimization problem a Lagrange multiplier λ is introduced:

$$L(\boldsymbol{u_1}, \lambda) = \boldsymbol{u_1}^T S \boldsymbol{u_1} - \lambda(\boldsymbol{u_1}^T \boldsymbol{u} - \mathbf{1})$$

Review of Lagrange Multiplier

Lagrange multipliers are used to find the maximum or minimum of a function $f(x, y)$ subject to constraint $g(x, y) = c$



"The red line shows the constraint g(x,y) = c. The blue lines are contours of f(x,y). The point where the red line tangentially touches a blue contour is our solution." [Lagrange Multipliers, Wikipedia

we define a new constant λ called a Lagrange Multiplier and we form the Lagrangian,

$$L(x, y, \lambda) = f(x, y) - \lambda g(x, y)$$

If $f(x^*, y^*)$ is the max of $f(x, y)$, there exists λ^* such that (x^*, y^*, λ^*) is a stationary point of L (partial derivatives are 0). In addition (x^*, y^*) is a point in which functions f and g touch but do not cross. At this point, the tangents of f and g are parallel or gradients of f and g are parallel, such that:

$$\nabla_{x,y} f = \lambda \nabla_{x,y} g$$

where, $\nabla_{x,y} f = (\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}) \leftarrow$ the gradient of f

$$\nabla_{x,y} g = (\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}) \leftarrow \text{the gradient of } g$$

Example :

Suppose we want to maximize the function $f(x, y) = x - y$ subject to the constraint $x^2 + y^2 = 1$.

We can apply the Lagrange multiplier method to find the maximum value for the function f ; the Lagrangian is:

$$L(x, y, \lambda) = x - y - \lambda(x^2 + y^2 - 1)$$

We want the partial derivatives equal to zero:

$$\frac{\partial L}{\partial x} = 1 + 2\lambda x = 0$$

$$\frac{\partial L}{\partial y} = -1 + 2\lambda y = 0$$

$$\frac{\partial L}{\partial \lambda} = x^2 + y^2 - 1$$

Solving the system we obtain two stationary points: $(\sqrt{2}/2, -\sqrt{2}/2)$ and $(-\sqrt{2}/2, \sqrt{2}/2)$. In order to understand which one is the maximum, we just need to substitute it in $f(x, y)$ and see which one as the biggest value. In this case the maximum is $(\sqrt{2}/2, -\sqrt{2}/2)$.

$$L(\boldsymbol{u_1}, \lambda) = \boldsymbol{u_1}^T S \boldsymbol{u_1} - \lambda(\boldsymbol{u_1}^T \boldsymbol{u_1} - 1) \tag{1}$$

Differentiating with respect to $\boldsymbol{u_1}$ gives d equations,

$$S\boldsymbol{u_1} = \lambda \boldsymbol{u_1}$$

Premultiplying both sides by $\boldsymbol{u_1}^T$ we have:

$$\boldsymbol{u_1}^T S \boldsymbol{u_1} = \lambda \boldsymbol{u_1}^T \boldsymbol{u_1} = \lambda$$

$\boldsymbol{u_1}^T S \boldsymbol{u_1}$ is maximized if λ is the largest eigenvalue of S.

Clearly λ and $\boldsymbol{u_1}$ are an eigenvalue and an eigenvector of S. Differentiating (1) with respect to the Lagrange multiplier λ gives us back the constraint:

$$\boldsymbol{u_1}^T \boldsymbol{u_1} = 1$$

This shows that the first principal component is given by the eigenvector with the largest associated eigenvalue of the sample covariance matrix S. A similar argument can show that the p dominant eigenvectors of covariance matrix S determine the first p principal components.