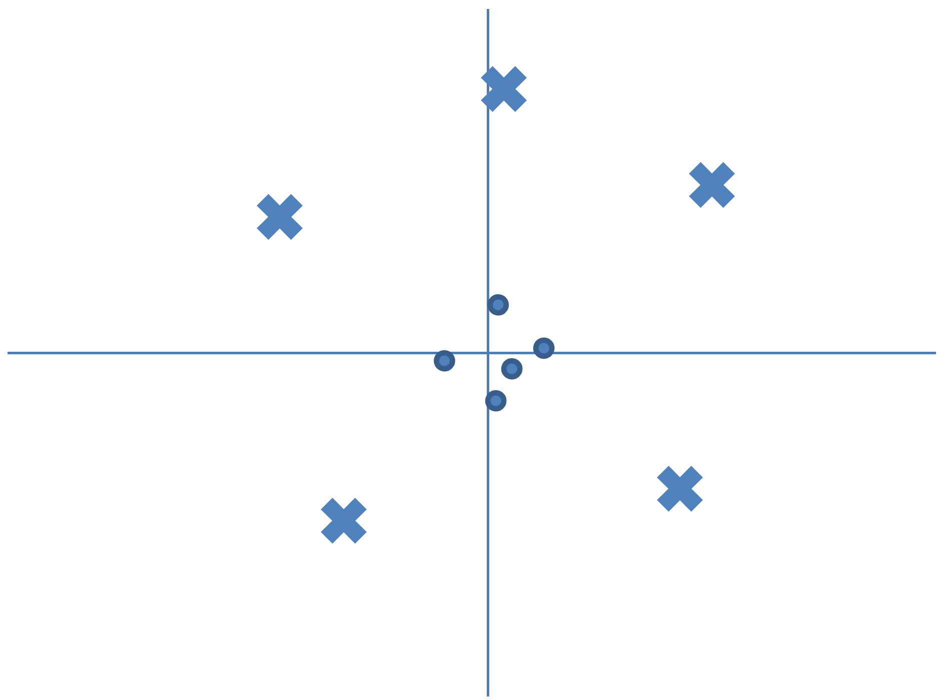
Kernels and kernel nearest means

Usman Roshan

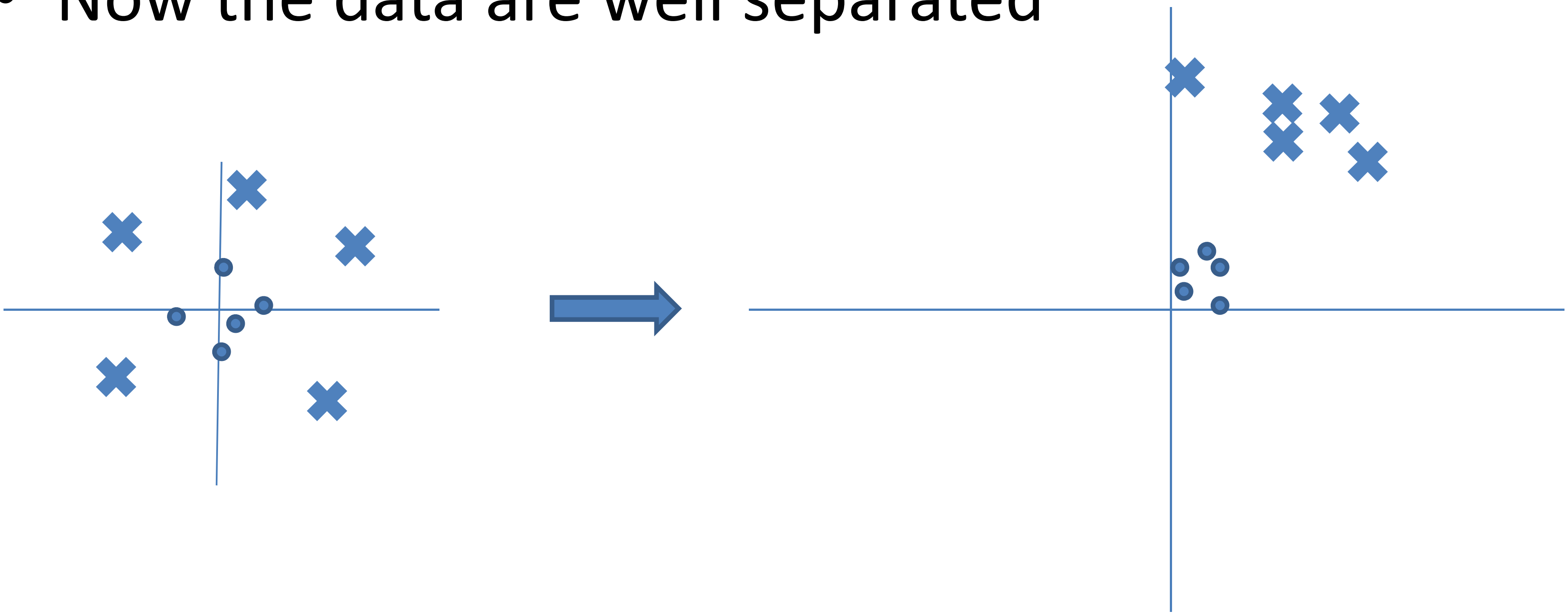
Feature space representation

- Consider two classes shown below
- Data cannot be separated by a hyperplane



Feature space representation

- Suppose we square each coordinate
- In other words $(x_1, x_2) \Rightarrow (x_1^2, x_2^2)$
- Now the data are well separated



Feature spaces/Kernel trick

- Using a linear classifier (nearest means or SVM) we solve a non-linear problem simply by working in a different feature space.
- With kernels
  - we don't have to make the new feature space explicit.
  - we can implicitly work in a different space and efficiently compute dot products there.

Computing Euclidean distances in a different feature space

- The advantage of kernels is that we can compute Euclidean and other distances in different features spaces without explicitly doing the feature space conversion.

Computing Euclidean distances in a different feature space

- First note that the Euclidean distance between two vectors can be written as

$$\|x-y\|^2 = (x-y)^T(x-y) = x^T x + y^T y - 2x^T y$$

- In feature space we have

$$\begin{aligned}\|\phi(x)-\phi(y)\|^2 &= (\phi(x)-\phi(y))^T(\phi(x)-\phi(y)) \\ &= \phi(x)^T\phi(x)+\phi(y)^T\phi(y)-2\phi(x)^T\phi(y) \\ &= K(x,x)+K(y,y)-2K(x,y)\end{aligned}$$

where K is the kernel matrix.

Computing distance to mean in feature space

- Recall that the mean of a class (say C_1) is given by

$$
\begin{aligned}
m_1 &= \left(1/n_1 \right) x_1 + x_2 + \cdots + x_{n_1} \\
&= \left(1/n_1 \right) \sum_{i=1 \cdots n_1} x_1
\end{aligned}
$$

- In feature space the mean Φ_m would be

$$
\phi_m = \left(1/n_1 \right) \phi(x_1) + \phi(x_2) + \cdots + \phi(x_{n_1}) = \left(1/n_1 \right) \sum_{i=1 \cdots n_1} \phi(x_1)
$$

Computing distance to mean in feature space

$$
\begin{aligned}
K(m,m) &= \phi_m^T \phi_m \\
&= \left(\left(\tfrac{1}{n} \right) \sum_{i=1 \ldots n} \phi(x_i) \right)^T \left(\left(\tfrac{1}{n} \right) \sum_{j=1 \ldots n} \phi(x_j) \right) \\
&= \tfrac{1}{n^2} \sum_{i=1 \ldots n} \sum_{j=1 \ldots n} \phi(x_i)^T \phi(x_j) \\
&= \tfrac{1}{n^2} \sum_{i=1 \ldots n} \sum_{j=1 \ldots n} K(x_i, x_j)
\end{aligned}
$$

Computing distance to mean in feature space

$$
\begin{aligned}
K(x,m) &= \phi_x^T \phi_m \\
&= \phi(x)^T \left(\left(1/n \right) \sum_{i=1\ldots n} \phi(x_i) \right) \\
&= 1/n \sum_{i=1\ldots n} \phi(x)^T \phi(x_i) \\
&= 1/n \sum_{i=1\ldots n} K(x, x_j)
\end{aligned}
$$

Computing distance to mean in feature space

- Replace K(m,m) and K(m,x) with calculations from previous slides

$$\left\|\phi(m)-\phi(x)\right\|^2 = K(m,m)+K(x,x)-2K(m,x)$$

$$= \frac{1}{n^2}\sum_{i=1\ldots n}\sum_{j=1\ldots n} K(x_i,x_j)+K(x,x)-2\frac{1}{n}\sum_{i=1\ldots n} K(x,x_j)$$

Kernel nearest means algorithm

- Compute kernel
- Let x_i ($i=0..n-1$) be the training datapoints and y_i ($i=0..n'-1$) the test.
- For each mean mi compute $K(m_i,m_i)$
- For each datapoint y_i in the test set do
  - For each mean m_j do
    - $d_j = K(m_j,m_j) + K(y_i,y_i) - 2K(m_i,y_j)$
    - Assign y_i to the class with the minimum d_j