Linear Data Analysis
Nonlinear Separation

Cain Susko

Queen's University
School of Computing

March 29, 2022

a High Dimensional PCA

the main concept of this lecture is how one can perform a PCA on higher dimension data as well as how to avoid directly embedding vectors using the Gram matrix. It will also cover the derivation and algorithm for kernel PCA.

b Scatter Matrix of Observations

Given the data: $A \in \mathbb{R}^{m \times n}$, we can find the zero mean matrix as:

$$\begin{aligned} M &= A - \vec{1}\bar{A} \\ &= [I - \frac{1}{m}\vec{1}\vec{1}^{\top}]A \\ &= G_m A \end{aligned}$$

Where I is the identity matrix and G is called the centring.

Style (1) Therefore, the scatter matrix is:

$$S_v = M^{\top}M$$

If we then write M as $M = V\Sigma^{\top}\Sigma V^{\top}$, then S_v equals:

$$S_v = V\Lambda_v V^{\top}$$

Style (2) Given: $S_u = MM^{\top}$, we can expand to:

$$S_u = U\Sigma\Sigma^{\top}U^{\top}$$

which simplifies to be:

$$S_u = U\Lambda_u U^{\top}$$

Scoring

The above examples show the column (1) and row (2) form of the scatter matrix. Because the rank of M equals r, this implies the first r eigenvalues in the variable style Λ_v and observation style Λ_u are equal !!! Therefore, in order to score either style one can use a single equation:

$$Z = U\Lambda^{\frac{1}{2}}$$

We can also rewrite S_u using the centring matrix such that:

$$\begin{aligned} S_u &= MM^\top \\ &= G_m A A^\top G_m^\top \end{aligned}$$

c Kernel PCA Using The Gram Matrix

Given m observations, embed $\underline{a}_j \hookrightarrow \hat{a}_j$.

$$\hat{S}_u \in \mathbb{R}^{P \times P}$$

Recall, the Gram matrix $\hat{W} \in \mathbb{R}$ is square, symmetric, and positive semi-definite. It's entries are:

$$\hat{W}_{i,j} =^{def} k(\underline{a}_i, \underline{a}_j)$$

Therefore, the scatter matrix would be:

$$\begin{aligned} \hat{S}_u &= G_m[\hat{A}\hat{A}^\top]G_m^\top \\ &= G_m \hat{W} G_m^\top \end{aligned}$$

This means we will never need to embed vectors, instead we will compute each entry of $\hat{W}$ using the kernel function.

Kernel PCA

after computing $\hat{S}_u = \hat{U}\hat{\Lambda}\hat{U}^\top$ (notation changed for ease of understanding), we then perform PCA which we can use to find the score:

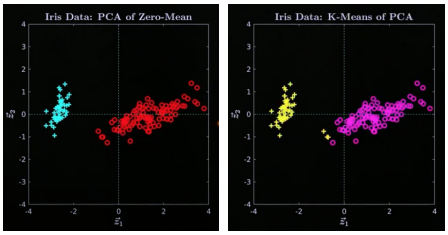
$$\hat{Z} = \hat{U}\hat{\Sigma}$$

This kernel PCA is slower than PCA as there are normally many more observations than variables.

d Kernel PCA on Iris Data

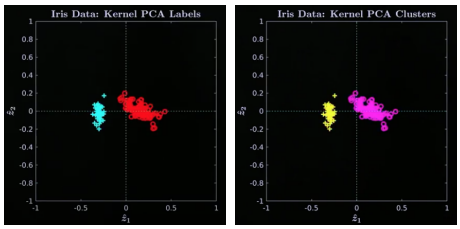
There will be 2 examples; One for conventional PCA and one for Kernel PCA. They will both use k-means to cluster and the kernel being used in kernel PCA is Gaussian with $\sigma^2 = m = 150$

Conventional after reducing and scoring the data the kmeans plot is:



Which is an ok–but not great seperation (outliers)

Kernel Kernel PCA resulted in:



which is a much more accurate separation of the data.

Learning Outcomes

Students should now be able to:

- compute a Gram matrix from given data
- using the centring matrix to center Gram matrix for PCA
- compute scores for Kernel PCA
- assess the results for simple data