CS281B/Stat241B: Advanced Topics in Learning & Decision Making

Kernel Principal Component Analysis

Lecturer: Michael I. Jordan *Scribes: Kamin Whitehouse*

First, we will describe the standard version of Principal Component Analysis (PCA). Then, we will describe Kernel PCA.

1 Standard PCA

We are given a data set $X = x_i$ for $i = 1 : m$ which is centered around the origin, i.e. $\sum_{i=1}^{m} x_i = 0$.

The sample covariance matrix C of this data is defined as $C = \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T$.

One possible data set $X \epsilon R^2$ is illustrated in Figure 1. The object of PCA is to replace the standard x and y axes with the first and second principal components of the data, which will call $\hat{x}$, $\hat{y}$, by redefining the data in terms of a $(\hat{x}, \hat{y})$ coordinate system. We can choose the first component according to one of two equivalent criteria

1. The projections of the data onto the first component should have maximal variance
2. The 'distance' from the data to the first component should be minimal. (Note that this is different from regression, in which we measure distance along the y axis. In this case, we measure the distance orthogonal to the first component, i.e. along the $\hat{y}$ axis).

All subsequent components can likewise be defined as, e.g. an orthogonal direction with the next largest variance, etc.

Here, we will show that the first criteria is equivalent to finding the eigenvectors of the sample covariance matrix C. The same is true for the second criteria, but a proof of this is left as an excercise.

1.1 proof

Let $y_i = w^T x_i$, where w will be our first eigenvector of C. That makes y_i the projection of x_i on w, our first principal component of x_i.

Assuming the data is centered around the origin, the sum of the squares of these projections will be their variance. So we want to maximize $\sum_i y_i^2$ subject to $w^T w = 1$. First, we can redefine this as:

$$\sum_i y_i^2 = \sum_i (w^T x_i)(x_i^T w) \tag{1}$$

$$\sum_i y_i^2 = w^T (\sum_i x_i x_i^T) w \tag{2}$$

$$\sum_i y_i^2 = m w^T C w \tag{3}$$

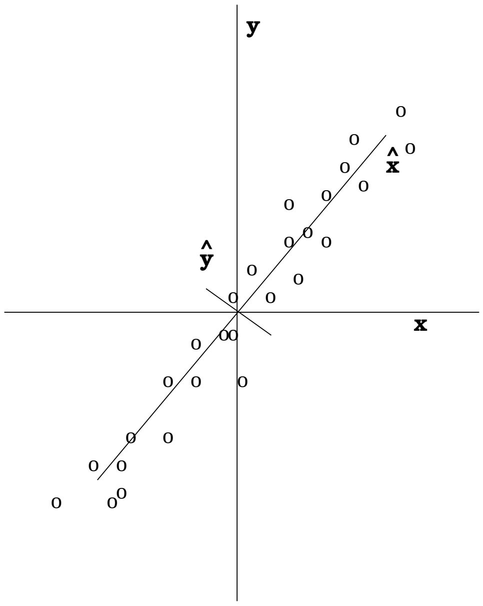

Figure 1

by the definition of C. Second, we note that this is a standard optimization problem, and we define the Lagrangian as follows:

$$\ell = mw^T Cw - \lambda(w^T w - 1)$$

We want to maximize this over w, so we take the partial derivative and set it equal to zero as follows

$$2mCw - 2\lambda w = 0 \tag{4}$$

$$Cw = \frac{\lambda}{m} w \tag{5}$$

Now we have shown that the maximal projection criteria is equivalent to an eigenvector problem. Here, we want to maximize λ to choose the first principal component. The second largest λ will give us the second principal component, etc.

2 Kernel PCA

Standard PCA is used to find a new set of axes (the principal components of the data) which hopefully better describes the data for some particular purpose. Here are two examples of why we might want to do this

1. If our data lies mainly within some hyperplane of R^m of lesser dimensionality than our current space, R^n, we can use only the first m principal components of our data to lower the dimensionality of our data from $n \to m$ without losing too much information (in fact we may only lose noise). This is the case in figure 1.

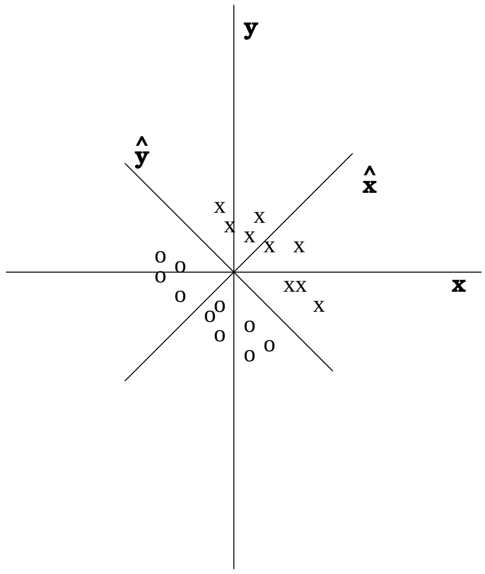

Figure 2

2. If our data are not clearly separable in any particular axis of our given space, we may be able to redefine the axes such that they are. This is illustrated in figure 2: the data is not separable on either x or y, but it is separable on $\hat{x}$.

But what happens when the data does not fall neatly into some hyperplane, as illustrated in figure 3? Or when our data falls into more than n clusters, where n is the dimensionality of the data (i.e. we don't have enough principal components to make all the data separable), as in figure 4? If we could define our principal components to be arbitrary manifolds we could get higher variance and better separability.

If we first map our data into a higher dimensional space where the data falls neatly onto some hyperplane or where the data is separable (or whatever our criteria may be) we can perform Standard PCA in that higher dimensional space. These principal components will map into the arbitrary manifolds that we want in our lower dimensional space. Kernel PCA allows us to perform PCA in this higher dimension while, using the kernel trick, doing all our calculations in a lower dimension.

2.1 The Kernel PCA method

First, we choose a transformation $\phi(x)$ $R^m \rightarrow R^n$, $m < n$. We apply these transformations to the data and arrive at a new sample covariance matrix $C = \frac{1}{m}\sum_i \phi(x_i)\phi(x_i)^T$, noting that our transformation must conserve our initial assumption that the data is centered over the origin.

To find the first principal component, we need to solve $Cv = \lambda v$. Substituting for C, this equation becomes

$$\frac{1}{m}\sum_i \phi(x_i)(\phi^T(x_i)v) = \lambda v \tag{6}$$

This is the equation we want to solve, but we want to try not to have to do it in the higher-dimensional space. Here, we make a key insight: since $\lambda \neq 0$, v must be in the span of $\phi(x_i)$, i.e. v can be written as some linear combination of $\phi(x_i)$. In other words, there must exist some set of α_i such that $v = \sum_i \alpha_i \phi(x_i)$ So, substituting for v we have

$$\lambda \sum_i \alpha_i \phi(x_i) = \frac{1}{m} \sum_i \phi(x_i)\phi^T(x_i) \sum_i \alpha_i \phi(x_i) \tag{7}$$

At this point, not all transformations $\phi(x)$ are in an inner product. So now we use a trick: we take the inner product of this with $\phi(x_n)$, $n = 1 : m$. This gives us:

$$\lambda \sum_i \alpha_i \phi^T(x_n)\phi(x_i) = \frac{1}{m} \sum_j \alpha_j \sum_i (\phi^T(x_n)\phi(x_i))(\phi^T(x_i)\phi(x_j)) \tag{8}$$

Now, since all instances of $\phi(x)$ are contained in an inner product, we can substitute this inner product with a kernel function, thereby allowing us to solve the problem without working directly in a high-dimensional space.

To substitute the inner products for a kernel function, we define a matrix K called the Gram Matrix, where

$$K_{ij} = \phi^T(x_i)\phi(x_j) = K(x_i, x_j)$$

Using the definition of K, we can rewrite equations 8 as

$$\lambda K \alpha = \frac{1}{m} K^2 \alpha \tag{9}$$

And now we see that we simply have to solve an eigenvalue problem on the Gram Matrix. Note that this matrix, however, grows with the number of data points, whereas in standard PCA we only have to solve for the eigenvectors of a matrix that grows with the *dimensionality* of the data points.

Now, to extract a principal component we simply take

$$y_i = v^t x$$

for all i, where v is the eigenvector corresponding to that principal component, noting that

$$y_i = \sum_j \alpha_i(\phi^T(x_j)\phi(x_i)) = \sum_j \alpha_j K(x_i, x_j)$$

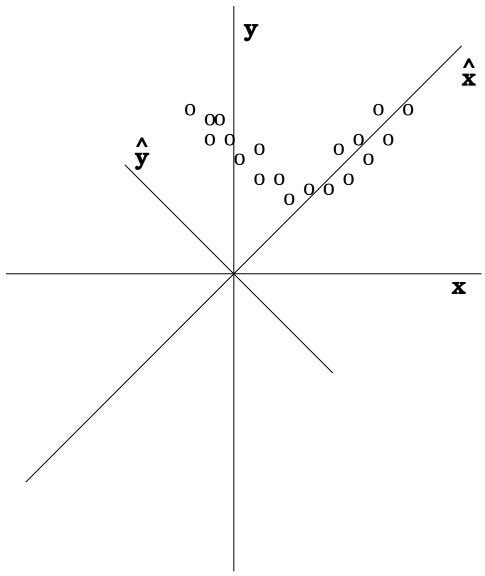

Figure 3

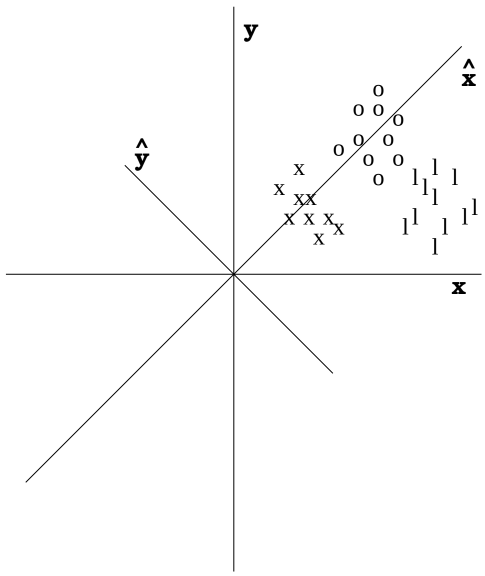

Figure 4