Extensions of Principal Component Analysis -Robust PCA and Kernel PCA

Darragh Spillane (S5270855)

November 2022

1 Introduction

Although principal component analysis is an extremely useful tool for reducing dimensionality of large data sets under some circumstances the assumptions made in order to effectively utilise it may not apply.Extending PCA to deal with these situations and to improve upon the standard method is a highly researched area and much progress has been made over the past few decades.Two of the main downfalls of PCA are its sensitivity to outliers and its assumption that features have a linear relationship.In this report we will explore two methods, namely Robust PCA(for dealing with outlier sensitivity)and Kernel PCA (for dealing with features with non linear relationships), which deal with these issues individually.It is assumed that the reader of this has a general understanding of standard PCA and its application however in the section on robust PCA two different outlooks of standard PCA (as a min/maximisation problem) are outlined in order to aid with the understanding of the rest of the material.

2 Robust PCA

In order to be able to fully comprehend Robust PCA it is important to understand how PCA can be seen as a minimisation problem and a maximisation problem.In the following sections there is a brief outline of how each problem can be represented which provides a good baseline for understanding PCA better as a whole and for understanding any extensions of standard PCA.

2.1 PCA as a Maximisation Problem

Suppose X is out data matrix (which has been centred) $\mathbf{c}$ is a principal component and let $\mathbf{w} = X\mathbf{c}$ be the projection of each data point on to the principal component $\mathbf{c}$. Since X has been centered the variance of the data after being projected on to the principal component z can be written as:

$$\mathbf{w}^T\mathbf{w} = \mathbf{c}^T X^T X\mathbf{c} \tag{1}$$

The objective of standard PCA is to maximise the variance after projecting on to the principal components so we can the problem as below.

$$\max_{\mathbf{c}}\{\mathbf{z}^T X^T X\mathbf{c} \mid \mathbf{c}^T\mathbf{c} = 1\} \tag{2}$$

Note: the extra condition that $\mathbf{c}^T\mathbf{c} = 1$ as we require the new basis to be orthonormal.

This Problem is then solved in the standard procedure of PCA by finding the eigenvectors of X^TX and setting $\mathbf{z}$ to the eigenvector with associated with the largest eigenvalue.

2.2 PCA as a Minimisation Problem

For the following consider $C = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k\}$ which is a matrix of the top k principal components and $W = XC$. Our approximation for the full data matrix X based on the top k principal components will be:

$$X \approx WC^T \tag{3}$$

We can now view the problem as a minimisation problem where we are trying to minimize the reconstruction error between the full data matrix X and our approximation WC^T i.e.

$$\min_C\{||X - WC^T||_F^2 : \; C^TC = I\} \tag{4}$$

where $||\dot{}||_F^2$ is the Frobenius Norm given by:

$$||X - WC^T||_F^2 = \sum_{j=1}^{n} ||X_j - WC_j^T||^2 \tag{5}$$

This minimisation problem is equivalent to maximising the variance. To see this consider a data point a_i (row i of X) then the contribution of that datapoint to the variance is $a_i^T a_i$, or equivalently the squared euclidean length $||a_i||_2^2$ Applying the Pythagorean theorem shows that this total variance equals the sum of variance lost (the squared residual) and variance remaining. Thus, it is equivalent to either maximize remaining variance or minimize lost variance to find the principal components. The figure below visualizes this for 2 dimensions:

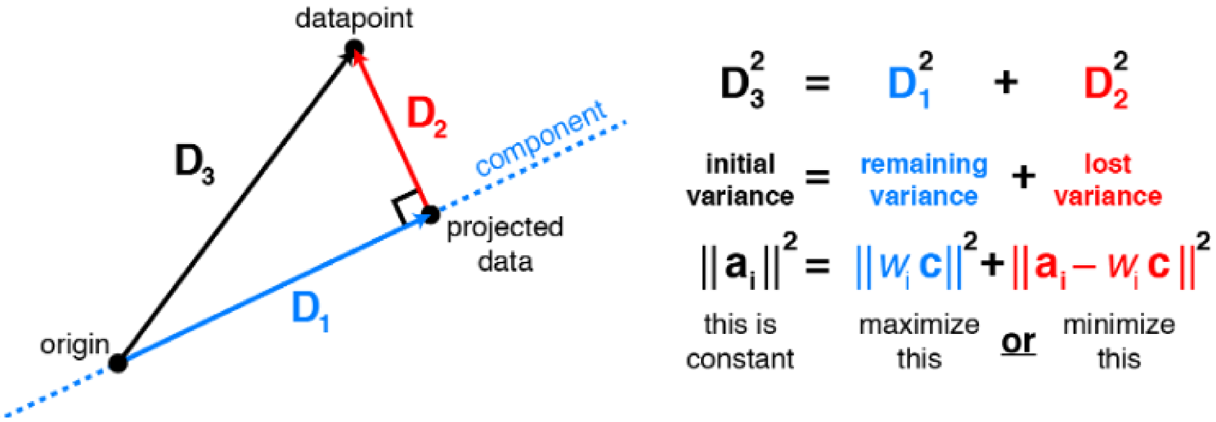


Figure 1: Maximizing variance in principal component space is equivalent to minimizing least-squares reconstruction error

2.3 Robust PCA

One of the downfalls of classic PCA is its sensitivity to outliers.This is due to the fact that PCA minimizes the L2 norm (as seen in the previous section) Because of the squaring of deviations from the outliers, they will dominate the total norm and therefore will drive the PCA components.See below an illustration of how the introduction of an outlier (in red) can effect the principal component axis of a data set.

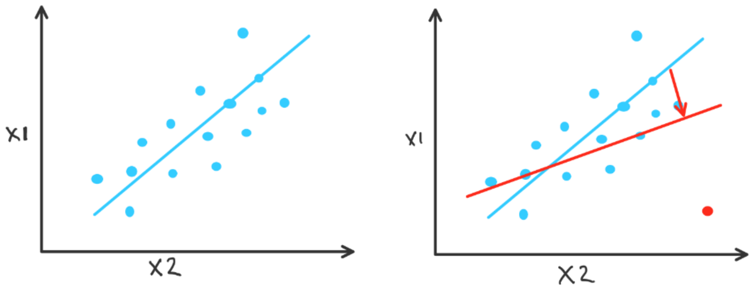


Figure 2: Outlines the effects of outlier on standard PCA.

The goal of Robust PCA is to reduce the effect of outliers.This is done by decomposing the original NxP data matrix X into two NxP matrices L and S. Where X = L+S and

- **L** - Is a low rank matrix which aims capture the trends across the data matrix.
- **S** - Is a sparse matrix which aims to capture the outlier measurements that obscure the low rank trends.

The main problem for Robust PCA is to find L and S such that the rank of L is minimized and that S is sparse i.e

$$\min_{L,S}\{rank(L) + ||S||_0\} \tag{6}$$

However this is a highly non convex problem and may not have a solution so the following minimization is used as a proxy.

$$\min_{L,S}\{||L||_* + a||S||_1\} \tag{7}$$

Where $|| \bullet ||_*$ is the nuclear norm.and $|| \bullet ||_1$ is the L1 norm.

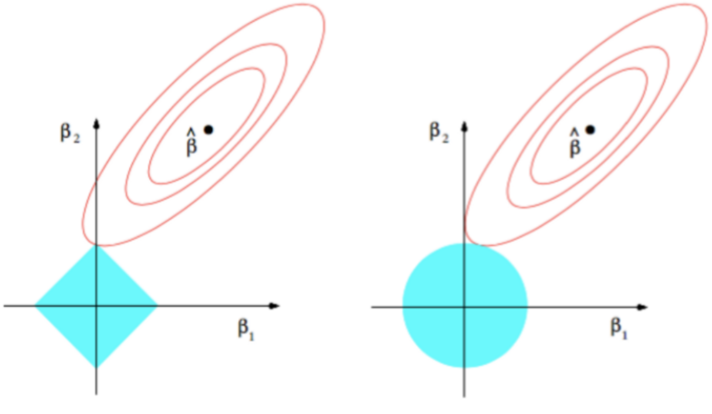
The Nuclear norm is defined as:

$$||A||_* = \sum_i d_i(A) \tag{8}$$

where $d_i(A)$ is the i-th singular value of the matrix A.

Since the number of non zero singular values is the rank minimizing the nuclear norm will yield a low rank matrix.

As seen in LASSO regression minimizing the L1 norm promotes sparsity. The reason for using the L1 norm to find a sparse solution is due to its special shape. It has spikes that happen to be at sparse points. Using it to touch the solution surface will very likely to find a touch point on a spike tip and thus a sparse solution.Although it is possible for a non sparse solution to be achieved using the L1 norm is far less likely compared to when using other norms.See below the very common illustration showing why the L1 norm tends to find sparse solutions compared to the L2 norm.(a useful graphic show in more detail how the below works can be found in this article - Why L1 norm creates Sparsity compared with L2 norm



Contours of the error and constraint functions.

Figure 3: The $\hat{\beta}$ is the unconstrained least squares estimate, the red ellipses are the contours of the least-squares error function and the blue regions are the constraint regions for L1 and L2 norm respectively

The problem of solving the (convex) minimization problem in (7) is known as principal component pursuit and there are many different optimization techniques that can be used to solve the best of which to use depends on the context in which the analysis is being carried out.

In some settings, the sparse component S may represent unwanted outliers, e.g. corrupted measurements — we may wish to clean the data by removing the outliers and recovering the low-rank component L.We can multiply (L+S) by C a compression matrix which we can tailor to remove/keep features as we please.

2.4 Examples and Applications of Robust PCA

Facial Recognition Robust PCA is often used when dealing with images and in image classification.Below is an example where robust PCA is used on an image of a face with a fake mustache.The image is split in to the low rank and sparse matrices where the low rank matrix represents a base which is (somewhat) common amongst all images (in the example below this is just the face without the fake mustache) and the sparse matrix reperesents some features(the fake mustache) which are not common and considered outliers.

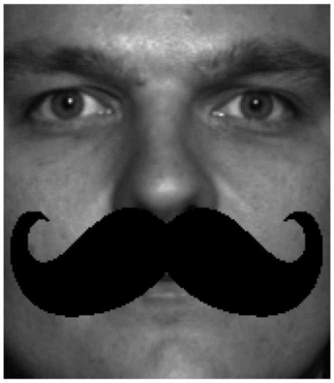


Figure 4: Image to be deconstructed with Robust PCA

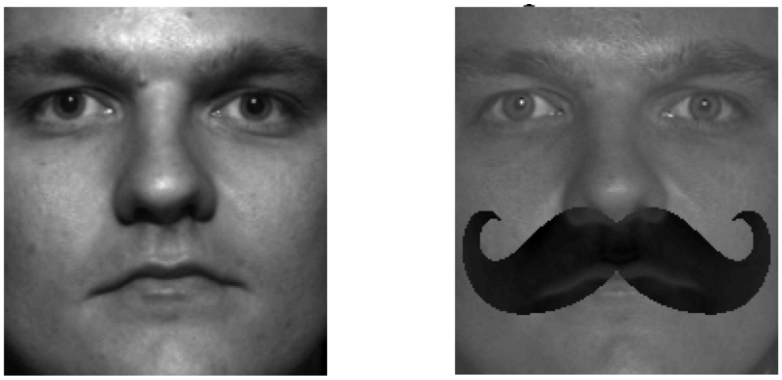


Figure 5: Decomposition where the left image represent L the low rank matrix matrix and the right represents S the Sparse matrix

It is clear to see from this example how useful Robust PCA can be for things like facial recognition where removing features such as glasses,piercings etc. are vital.

Fluid Dynamics:
Robust PCA is also being used in the area of fluid dynamics to separate the flow of a fluid in to base components and noise which not relevant to the problem.Below outliers have been added to a representation of a fluid flowing past an object and Robust PCA has been used to separate of the the image in to the actual flow and the outliers.

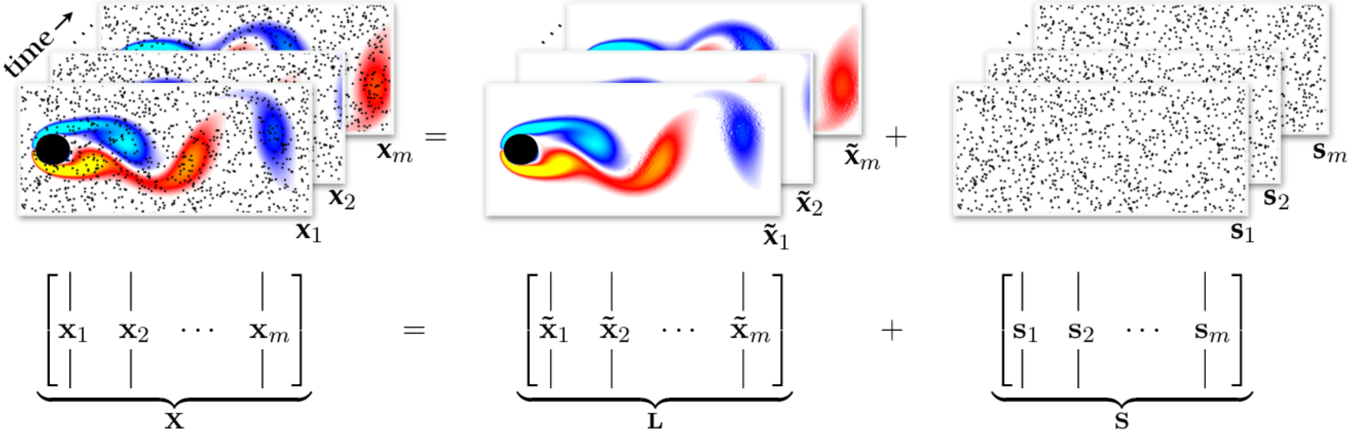

time
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_m$
=
$\tilde{\mathbf{x}}_1$
$\tilde{\mathbf{x}}_2$
$\tilde{\mathbf{x}}_m$
+
$\mathbf{s}_1$
$\mathbf{s}_2$
$\mathbf{s}_m$
$\underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} | & | & & | \\ \tilde{\mathbf{x}}_1 & \tilde{\mathbf{x}}_2 & \cdots & \tilde{\mathbf{x}}_m \\ | & | & & | \end{bmatrix}}_{\mathbf{L}} + \underbrace{\begin{bmatrix} | & | & & | \\ \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ | & | & & | \end{bmatrix}}_{\mathbf{S}}$

Figure 6: Outliers being removed in fluid dynamics.

3 Kernel PCA

Another one of PCA's downfalls is that it assumes linear relationship between features which may not always be the case and thus standard PCA will not always give us the optimal dimensionality reduction. To tackle non linear relationships between features we can use Kernel PCA (KPCA).Kernel PCA uses a kernel function to project dataset into a higher dimensional feature space, where it is linearly separable.

Suppose we have a set of N two dimensional data points $X = \{x^{(1)}, \dots x^{(N)}\}$ and suppose $\phi(x^{(i)})$ maps $x^{(i)}$ to a higher dimensional space.The covariance matrix C at the high dimensional ϕ-space can then be approximated by

$$C = \frac{1}{N}\sum_{i=1}^{N}\phi(x^{(i)})\phi(x^{(i)})^T \tag{9}$$

If we want to perform PCA in the higher dimensional space we will need to find $Cv = \lambda v$ which would require calculating the covariance matrix explicitly in the higher dimension which may be costly. To avoid explicitly computing the covariance matrix C by using the following Theorem

Theorem:
Any eigenvector of v can be represented as a weighted sum of $\phi(x^{(I)})$. That is

$$v = \sum_{i=1}^{N}\alpha_i\phi(x^{(i)}) \tag{10}$$

Subbing this into $Cv = \lambda v$ gives us

$$CV = \frac{1}{N}\sum_{i=1}^{N}\phi(x^{(i)})\phi(x^{(i)})^T\left(\sum_{j=1}^{N}\alpha_i\phi(x^{(j)})\right) \tag{11}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\phi(x^{(i)})\left(\sum_{j=1}^{N}\alpha_i\phi(x^{(i)})^T\phi(x^{(j)})\right) \tag{12}$$

Now define the Gram matrix G with it i,j element by

$$G_{i,j} = \langle\phi(x^{(i)}), \phi(x^{(j)})\rangle = \phi(x^{(i)})^T\phi(x^{(i)}) = K(x^{(i)}, x^{(j)}) \tag{13}$$

Where K is the Kernel function.

Now multiplying both sides of equation 12 by $\phi(x^{(k)})$ and doing some manipulation we get:

$$\lambda N\alpha = G\alpha \tag{14}$$

So α is actually an eigenvector of G with eigenvalue λN Similar to standard PCA, we sort the eigenvalues in descending order to select the principal components. And given an α the eigenvector in ϕ-space is

$$v = \sum_{i=1}^{N}\alpha_i\phi(x^{(i)}) \tag{15}$$

To project a new input x we can project v as

$$\langle\phi(x), v\rangle = \sum_{i=1}^{N}\alpha_i\langle\phi(x), \phi(x^{(i)})\rangle = \sum_{i=1}^{N}\alpha_i K(x, x^{(i)}) \tag{16}$$

Note: It is assumed in all of the above that the projected x in the ϕ-space are zero mean.

3.1 Method Outline:

- Choose a kernel K and compute the normalized Gram Matrix
  $G_{i,j} = K(x^{(i)}, x^{(j)})$
- Decompose the Gram matrix as you would the covariance matrix in standard PCA.
- Sort the eigenvalues in descending order with each eigenvector composed of the weights α which outline the make up of the principal components in the ϕ-space
- given an input x the projection to a principal component with weight α is given by:
  $\sum_{i=1}^{N} \alpha_i K(x, x^{(i)})$

3.2 Visualisation:

See below an example of a problem where the data is not linearly separable.Both PCA and KPCA have been applied to the data.Note PCA has little to no effect where as with KPCA the data points can be clearly separated.

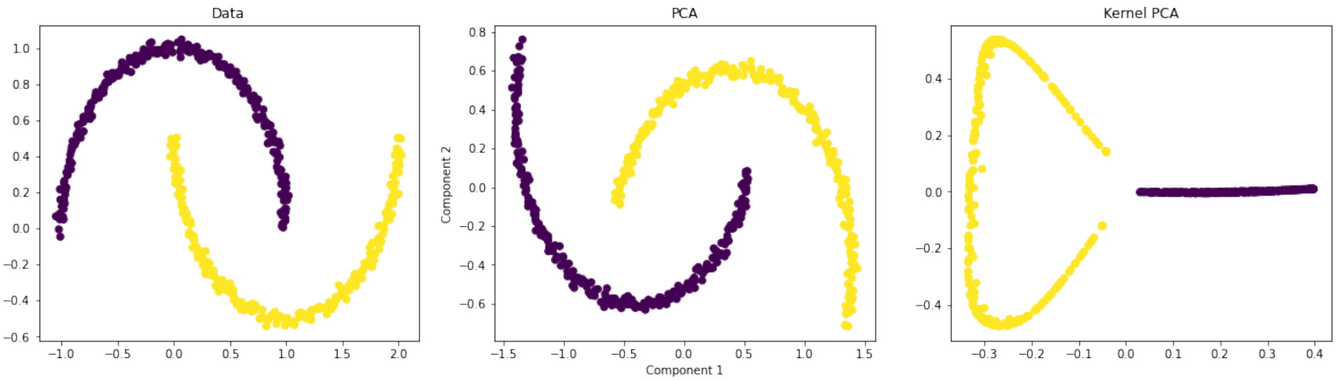


Figure 7: Performance of PCA and KPCA on a non linear data set.

4 Sources

http://alexhwilliams.info/itsneuronalblog/2016/03/27/pca/

https://iq.opengenus.org/kernal-principal-component-analysis/

https://royalsocietypublishing.org/doi/10.1098/rsta.2015.0202

https://medium.com/@aneetkumard8/understanding-robust-principal-component-analysis-rpca-d722aab80202

https://medium.com/@phsamuel.work/kernel-pca-f7a40f0b4265

https://satishkumarmoparthi.medium.com/why-l1-norm-creates-sparsity-compared-with-l2-norm-3c6fa9c607f4: :

https://www.geeksforgeeks.org/ml-introduction-to-kernel-pca/.