

PCA

Principal Component Analysis

Dr. Saed Sayad

University of Toronto

2010

saed.sayad@utoronto.ca

Basic Statistics

$$\bar{X} = \frac{\sum X}{n}$$
$$Var(X) = \frac{\sum_{i=1}^n (X - \bar{X})^2}{n-1}$$

$$Covar(X, Y) = \frac{\sum_{i=1}^n (X - \bar{X})(Y - \bar{Y})}{n-1}$$

Statistics - Example

X	SumX	Count	Average	X-Xbar	(X-Xbar)^2	Sxx	Variance	SD
0	40	4	10	-10	100	208	69.33	8.33
8				-2	4			
12				2	4			
20				10	100			
Y	SumY	Count	Average	Y-Ybar	(Y-Ybar)^2	Syy	Variance	SD
8	40	4	10	-2	4	10	3.33	1.83
9				-1	1			
11				1	1			
12				2	4			
X	Y	Count	X-Xbar	Y-Ybar	(X-Xbar)(Y-Ybar)	Sxy	CoVariance	
0	8	4	-10	-2	20	44	14.67	
8	9		-2	-1	2			
12	11		2	1	2			
20	12		10	2	20			

Covariance Matrix

$$C = \begin{bmatrix} \text{cov}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\ \text{cov}(y, x) & \text{cov}(y, y) & \text{cov}(y, z) \\ \text{cov}(z, x) & \text{cov}(z, y) & \text{cov}(z, z) \end{bmatrix}$$

Matrix Algebra

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 11 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

Example of one non-eigenvector and one eigenvector

Matrix Algebra

$$2 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 24 \\ 16 \end{bmatrix} = 4 \times \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

Example of how a scaled eigenvector is still an eigenvector

Eigenvectors Properties

- Eigenvectors can only be found for **square** matrices and not every square matrix has eigenvectors.
- Given an $n \times n$ matrix that does have eigenvectors, there are n of them.
- Another property of eigenvectors is that even if we scale the vector by some amount before we multiply it, we still get the same multiple of it as a result. This is because if you scale a vector by some amount, all you are doing is ***making it longer not changing it's direction***.
- Lastly, all the eigenvectors of a matrix are ***perpendicular***. it means that you can express the data in terms of these perpendicular eigenvectors, instead of expressing them in terms of the x and y axes.

Standardized Eigenvectors

- We like to find the eigenvectors whose length is exactly one.
- This is because, the length of a vector doesn't affect whether it's an eigenvector or not, whereas the direction does.
- So, in order to keep eigenvectors standard, whenever we find an eigenvector we usually scale it to make it have a length of 1, so that all eigenvectors have the same length.

Standardized Eigenvectors

$$\begin{bmatrix} 3 \\ 2 \end{bmatrix} \quad \text{Eigenvector}$$

$$\sqrt{(3^2 + 2^2)} = \sqrt{13} \quad \text{Vector Length}$$

$$\begin{bmatrix} 3 \\ 2 \end{bmatrix} \div \sqrt{13} = \begin{bmatrix} 3 / \sqrt{13} \\ 2 / \sqrt{13} \end{bmatrix} \quad \text{Eigenvector with length of one}$$

Eigenvalues

- Eigenvalues is the amount by which the original vector was scaled after multiplication by the square matrix.
- 4 is the *eigenvalue* associated with that eigenvector in the example.
- No matter what multiple of the eigenvector we took before we multiplied it by the square matrix, we would always get 4 times the scaled vector.
- Eigenvectors and Eigenvalues always come in pairs.

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \times \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \times \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

PCA

- PCA is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences.
- Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analysing data.
- The other main advantage of PCA is that once you have found these patterns in the data, and you compress the data, ie. by reducing the number of dimensions, without much loss of information. This technique used in image compression.

PCA – Original and Adjusted Data

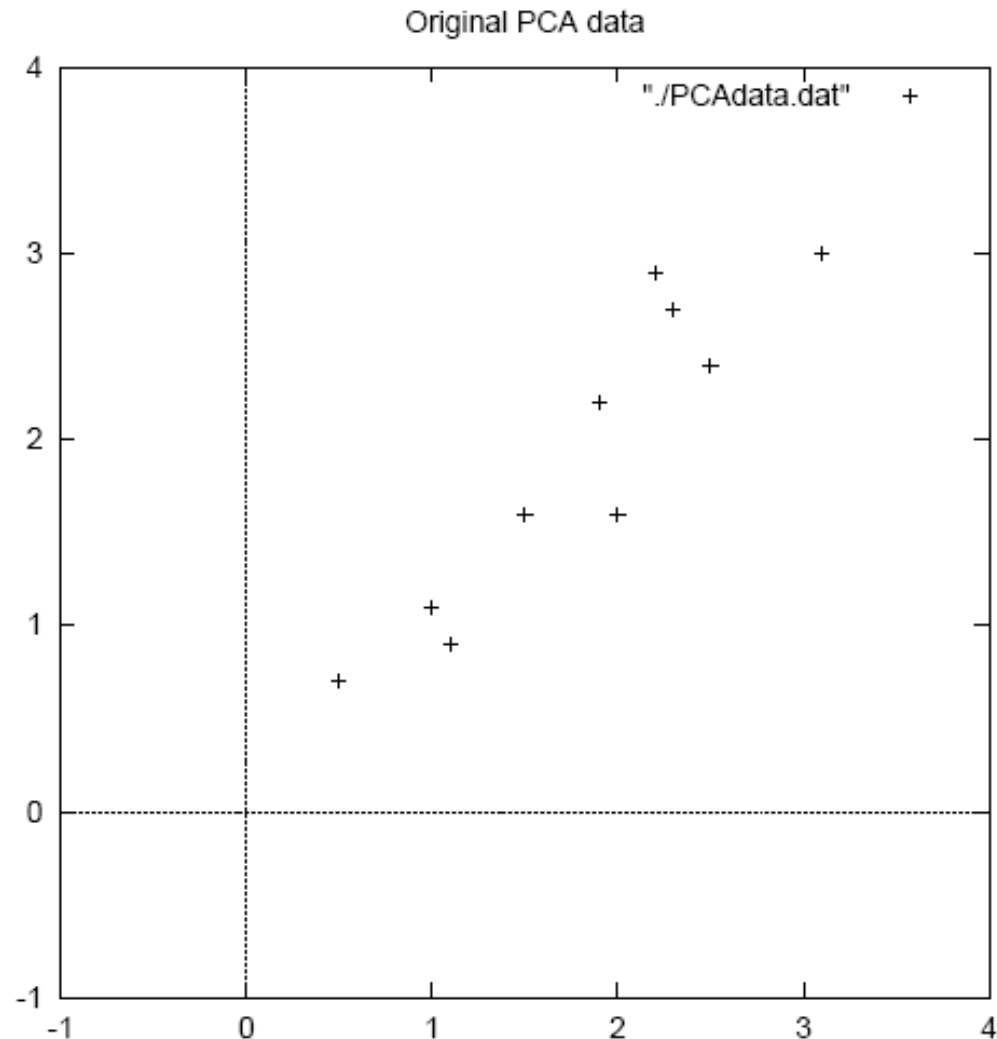
	x	y
Data =	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

Original Data

	x	y
DataAdjust =	.69	.49
	-1.31	-1.21
	.39	.99
	.09	.29
	1.29	1.09
	.49	.79
	.19	-.31
	-.81	-.81
	-.31	-.31
	-.71	-1.01

Original Data - Average

PCA – Original Data plot



Calculate Covariance Matrix

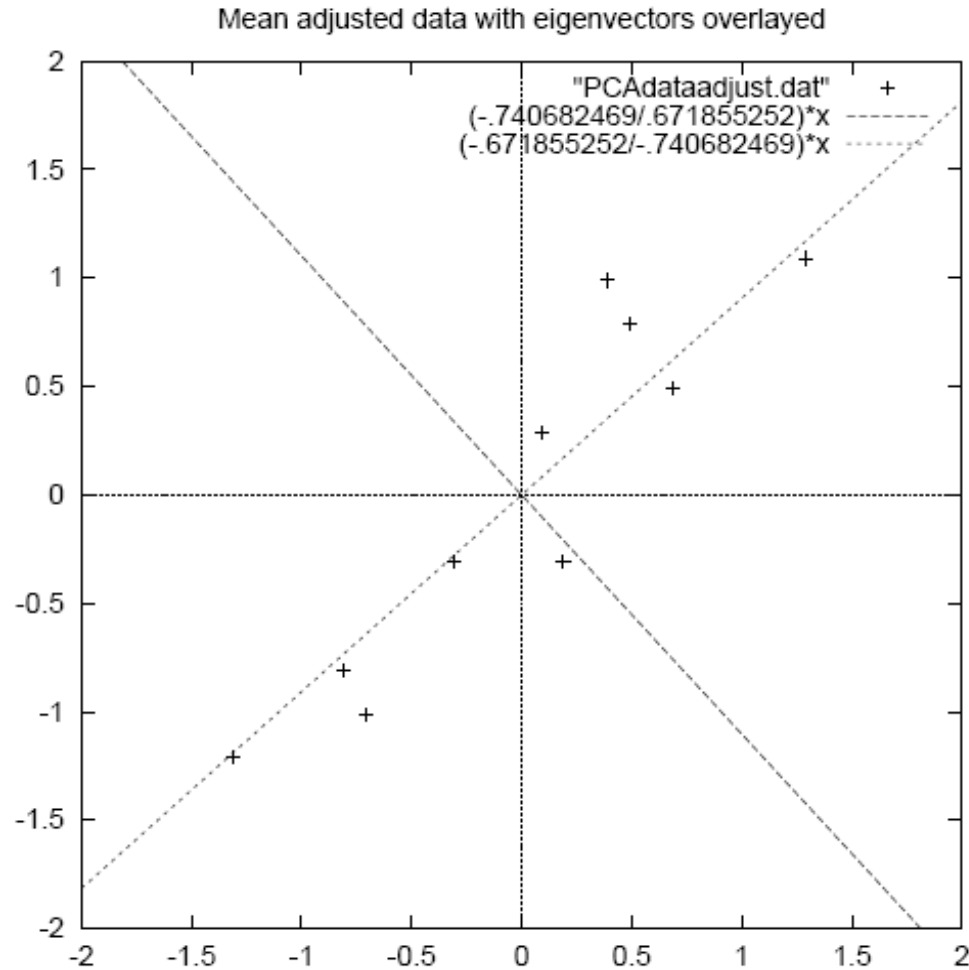
$$\text{cov} = \begin{bmatrix} 0.61656 & 0.61544 \\ 0.61544 & 0.71656 \end{bmatrix}$$

Calculate Eigenvectors and Eigenvalues from Covariance Matrix

$$\textit{eigenvalues} = \begin{bmatrix} 0.049083 \\ 1.284028 \end{bmatrix}$$

$$\textit{eigenvectors} = \begin{bmatrix} -0.73518 & -0.67787 \\ 0.67787 & -0.73518 \end{bmatrix}$$

Eigenvectors Plot



Choosing components and forming a Feature Vector

- The eigenvector with the **highest** eigenvalue is the **principle component** of the data set. It is the most significant relationship between the data dimensions.
- In general, once eigenvectors are found from the covariance matrix, the next step is to **order them by eigenvalue**, highest to lowest. This gives us the components in order of significance.
- Now, if we like, we can decide to **ignore** the components of lesser significance. We do lose some information, but if the eigenvalues are small, we don't lose much.
- If we leave out some components, the final data set will have less dimensions than the original.

Feature Vector

$$\textit{FeatureVector} = (eig_1, eig_2, eig_3, \dots, eig_n)$$

$$\textit{eigenvectors} = \begin{bmatrix} -0.67787 & -0.73518 \\ -0.73518 & 0.677873 \end{bmatrix}$$

$$\textit{pc} = \begin{bmatrix} -0.67787 \\ -0.73518 \end{bmatrix}$$

Deriving the new data set

$$\text{Final Data} = \text{Row Feature Vector} \times \text{Row Data Adjust}$$

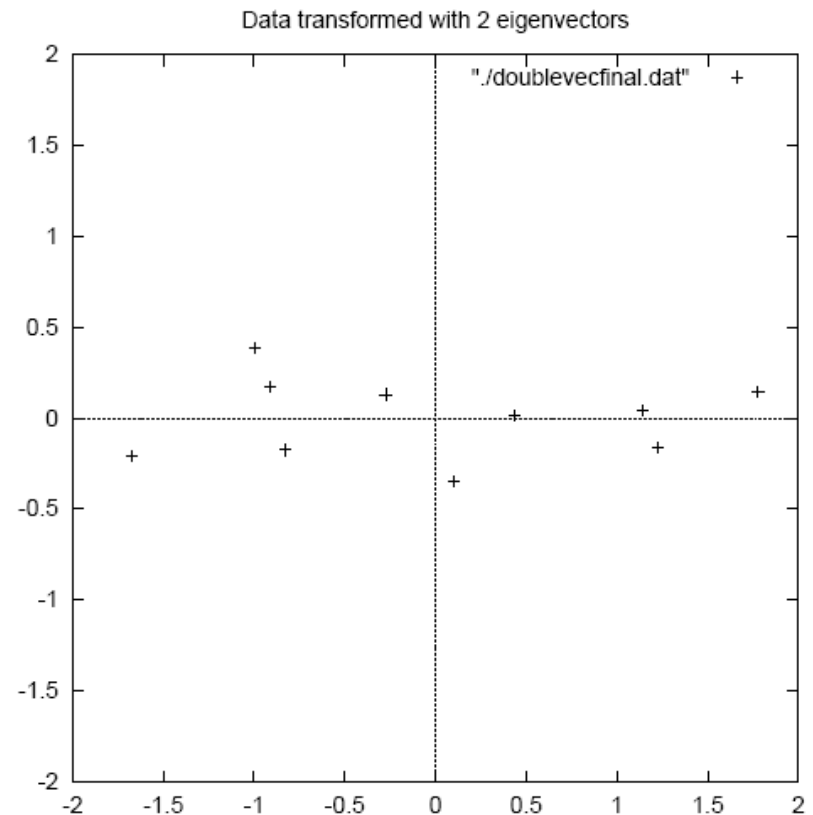
- **Row Feature Vector** is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top.
- **Row Data Adjust** is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.
- **Final Data** is the final data items in columns, and dimensions along rows. It is the original data ***solely in terms of the vectors***.

Deriving the new data set

Two Eigenvectors

Transformed Data=

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287



Get the original data back

$$\text{Row Data Adjust} = \text{Row Feature Vector}^T \times \text{Final Data}$$

$$\text{Row Original Data} = (\text{Row Feature Vector}^T \times \text{Final Data}) + \text{Original Mean}$$

Principal Component Regression

PCA + MLR

PCR

$$\mathbf{X} = \mathbf{T}\mathbf{P}$$

$$\mathbf{Y} = \mathbf{T}\mathbf{B} + \mathbf{E}$$

$$\mathbf{B} = (\mathbf{T}'\mathbf{T})^{-1}\mathbf{T}'\mathbf{Y}$$

- **T** is a matrix of SCORES
- **X** is a DATA matrix
- **P** is a matrix of LOADINGS
- **Y** is a dependent variable vector
- **B** is the regression coefficients vector

PCR

- In PCR the **X** matrix is replaced by the **T** matrix which has less and **orthogonal variables**.
- There will be no issue with **inversion** of the **T'T** matrix (unlike MLR) because of orthogonal scores.
- PCR also resolves the issue of **colinearity** which could in return reduce the prediction error.
- There is no guarantee to have a PCR model that works better than an MLR model using the same dataset.

Reference

- *www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf*



Questions?