

Dimensionality Reduction and Feature Construction

- Principal components analysis (PCA)
 - Reading: L. I. Smith, *A tutorial on principal components analysis* (on class website)
 - PCA used to reduce dimensions of data without much loss of information.
 - Used in machine learning and in signal processing and image compression (among other things).

PCA is “an orthogonal linear transformation that transfers the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (*first principal component*), the second greatest variance lies on the second coordinate (*second principal component*), and so on.”

Background for PCA

- Suppose attributes are A_1 and A_2 , and we have n training examples. x 's denote values of A_1 and y 's denote values of A_2 over the training examples.
- Variance of an attribute:

$$\text{var}(A_1) = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{(n-1)}$$

- Covariance of two attributes:

$$\text{cov}(A_1, A_2) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)}$$

- If covariance is positive, both dimensions increase together. If negative, as one increases, the other decreases. Zero: independent of each other.

- Covariance matrix
 - Suppose we have n attributes, A_1, \dots, A_n .
 - Covariance matrix:

$$C^{n \times n} = (c_{i,j}), \text{ where } c_{i,j} = \text{cov}(A_i, A_j)$$

	<i>Hours(H)</i>	<i>Muck(M)</i>
Data	9	39
	15	56
	25	93
	14	61
	10	50
	18	75
	0	32
	16	85
	5	42
	19	70
	16	66
	20	80
Totals	167	749
Averages	13.92	62.42

$$\begin{pmatrix} \text{cov}(H, H) & \text{cov}(H, M) \\ \text{cov}(M, H) & \text{cov}(M, M) \end{pmatrix}$$

$$= \begin{pmatrix} \text{var}(H) & 104.5 \\ 104.5 & \text{var}(M) \end{pmatrix}$$

Covariance:

<i>H</i>	<i>M</i>	$(H_i - H)$	$(M_i - M)$	$(H_i - H)(M_i - M)$
9	39	-4.92	-23.42	115.23
15	56	1.08	-6.42	-6.93
25	93	11.08	30.58	338.83
14	61	0.08	-1.42	-0.11
10	50	-3.92	-12.42	48.69
18	75	4.08	12.58	51.33
0	32	-13.92	-30.42	423.45
16	85	2.08	22.58	46.97
5	42	-8.92	-20.42	182.15
19	70	5.08	7.58	38.51
16	66	2.08	3.58	7.45
20	80	6.08	17.58	106.89
Total				1149.89
Average				104.54

$$= \begin{pmatrix} 47.7 & 104.5 \\ 104.5 & 370 \end{pmatrix}$$

Covariance matrix

Table 2.2: 2-dimensional data set and covariance calculation

- Eigenvectors:

- Let **M** be an $n \times n$ matrix.

- **v** is an *eigenvector* of **M** if $\mathbf{M} \times \mathbf{v} = \lambda \mathbf{v}$
- λ is called the *eigenvalue* associated with **v**

- For any eigenvector **v** of **M** and scalar a ,

- Thus you can always choose eigenvectors of length 1:

$$\mathbf{M} \times a\mathbf{v} = \lambda a\mathbf{v}$$

- If **M** has any eigenvectors, it has n of them, and they are orthogonal to one another.

- Thus eigenvectors can be used as a new basis for a n -dimensional vector space.

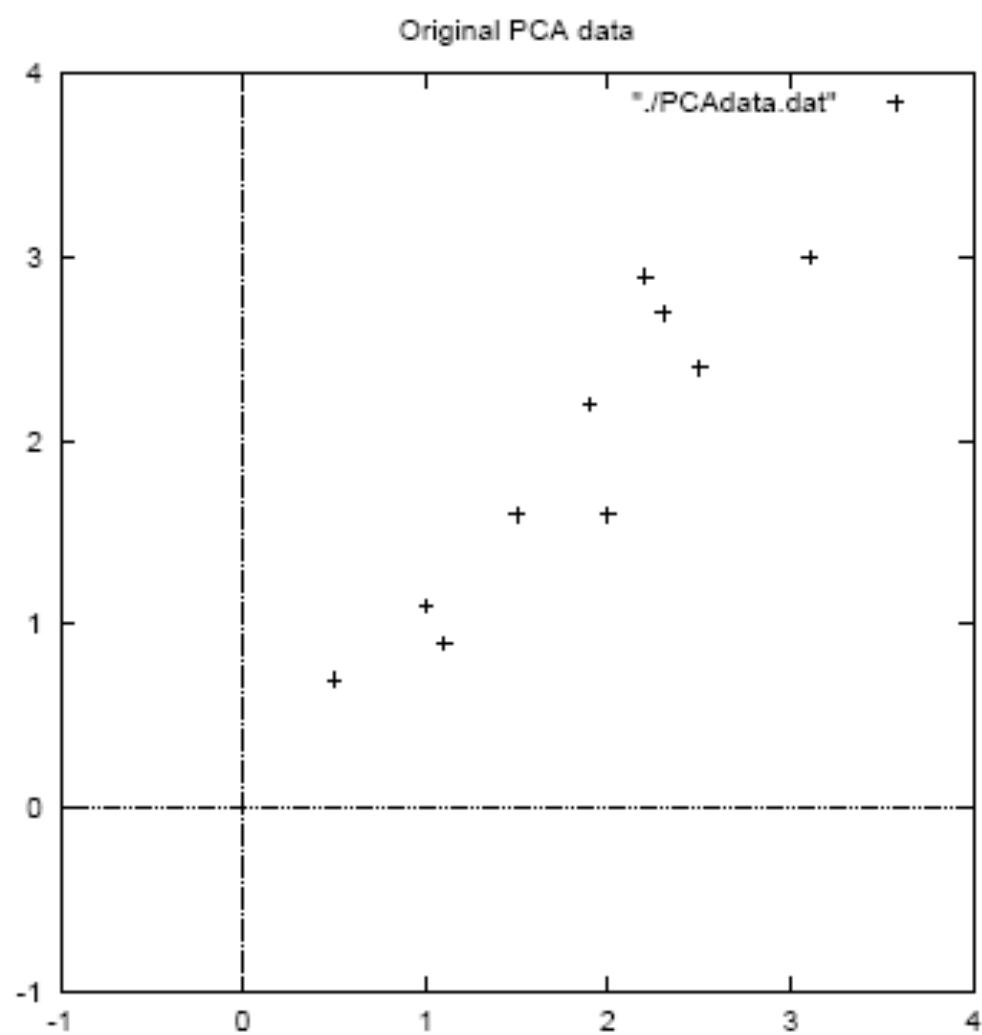
$$\sqrt{v_1^2 + \dots + v_n^2} = 1$$

PCA

1. Given original data set $S = \{\mathbf{x}^1, \dots, \mathbf{x}^k\}$, produce new set by subtracting the mean of attribute A_i from each x_i .

Data	x	y
	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
Mean: 1.81		1.91

DataAdjust	x	y
	.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
Mean: 0		0



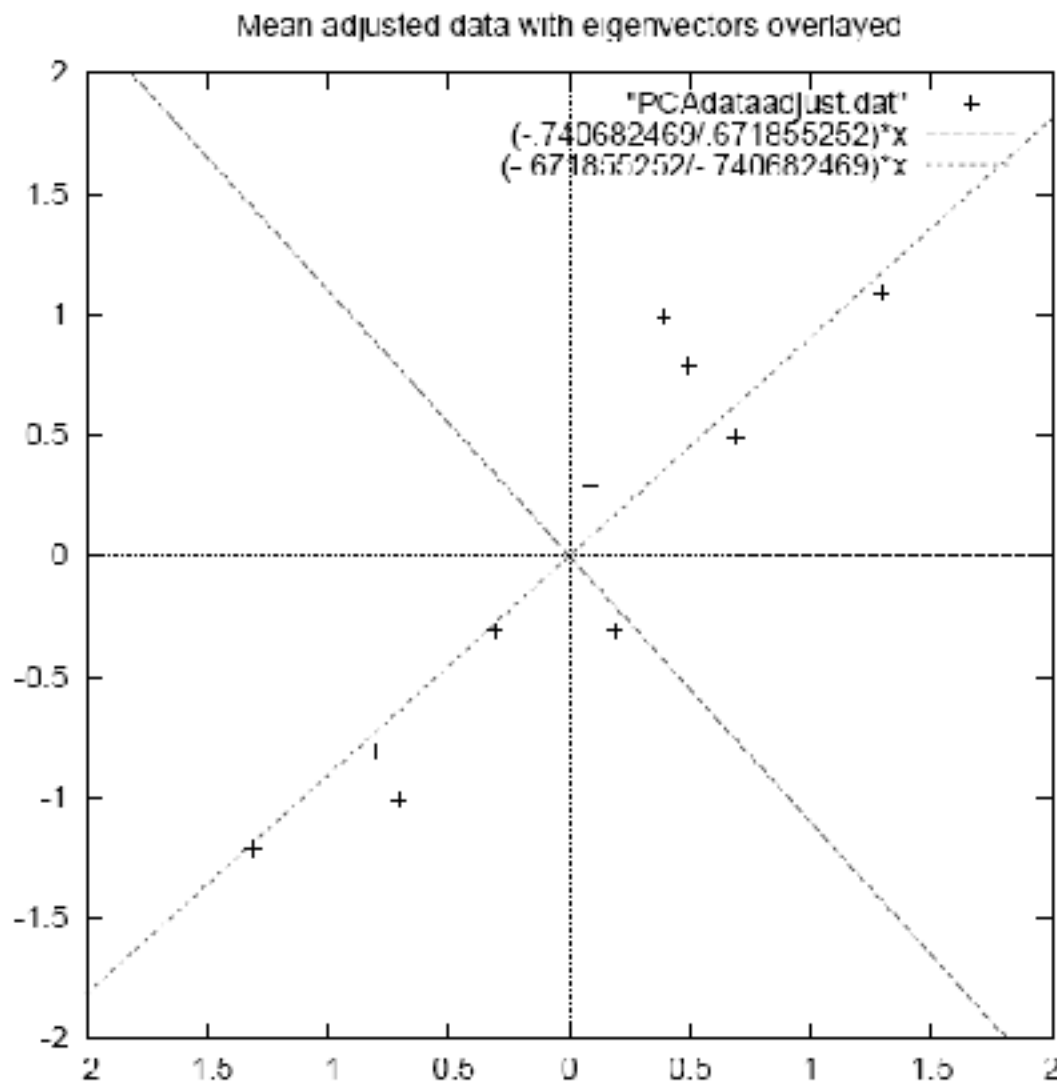
2. Calculate the covariance matrix:

$$\begin{matrix} & \begin{matrix} x & y \end{matrix} \\ \begin{matrix} cov \Rightarrow \end{matrix} & \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix} \end{matrix}$$

3. Calculate the (unit) eigenvectors and eigenvalues of the covariance matrix:

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$



Eigenvector with largest
eigenvalue traces
linear pattern in data

Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

4. Order eigenvectors by eigenvalue, highest to lowest.

$$\mathbf{v}_1 = \begin{pmatrix} -.677873399 \\ -.735178956 \end{pmatrix} \quad \lambda = 1.28402771$$

$$\mathbf{v}_2 = \begin{pmatrix} -.735178956 \\ .677873399 \end{pmatrix} \quad \lambda = .0490833989$$

In general, you get n components. To reduce dimensionality to p , ignore $n-p$ components at the bottom of the list.

Construct new feature vector.

Feature vector = $(\mathbf{v}_1, \mathbf{v}_2, \dots \mathbf{v}_p)$

$$FeatureVector1 = \begin{pmatrix} -.677873399 & -.735178956 \\ -.735178956 & .677873399 \end{pmatrix}$$

or reduced dimension feature vector :

$$FeatureVector2 = \begin{pmatrix} -.677873399 \\ -.735178956 \end{pmatrix}$$

5. Derive the new data set.

$$\textit{TransformedData} = \textit{RowFeatureVector} \times \textit{RowDataAdjust}$$

$$\textit{RowFeatureVector1} = \begin{pmatrix} -.677873399 & -.735178956 \\ -.735178956 & .677873399 \end{pmatrix}$$

$$\textit{RowFeatureVector2} = \begin{pmatrix} -.677873399 & -.735178956 \end{pmatrix}$$

This gives original data in terms of chosen components (eigenvectors)—that is, along these axes.

$$\textit{RowDataAdjust} = \begin{pmatrix} .69 & -1.31 & .39 & .09 & 1.29 & .49 & .19 & -.81 & -.31 & -.71 \\ .49 & -1.21 & .99 & .29 & 1.09 & .79 & -.31 & -.81 & -.31 & -1.01 \end{pmatrix}$$

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

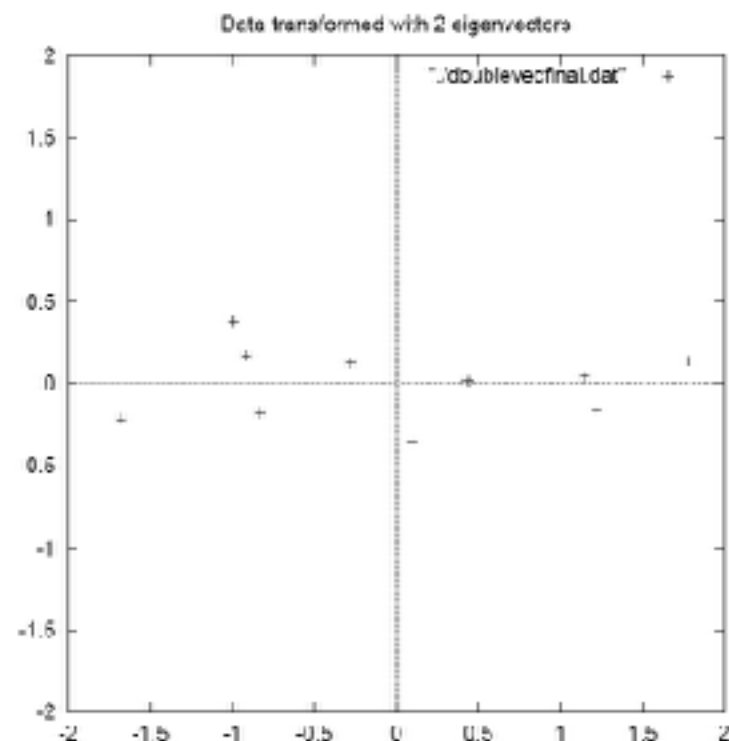
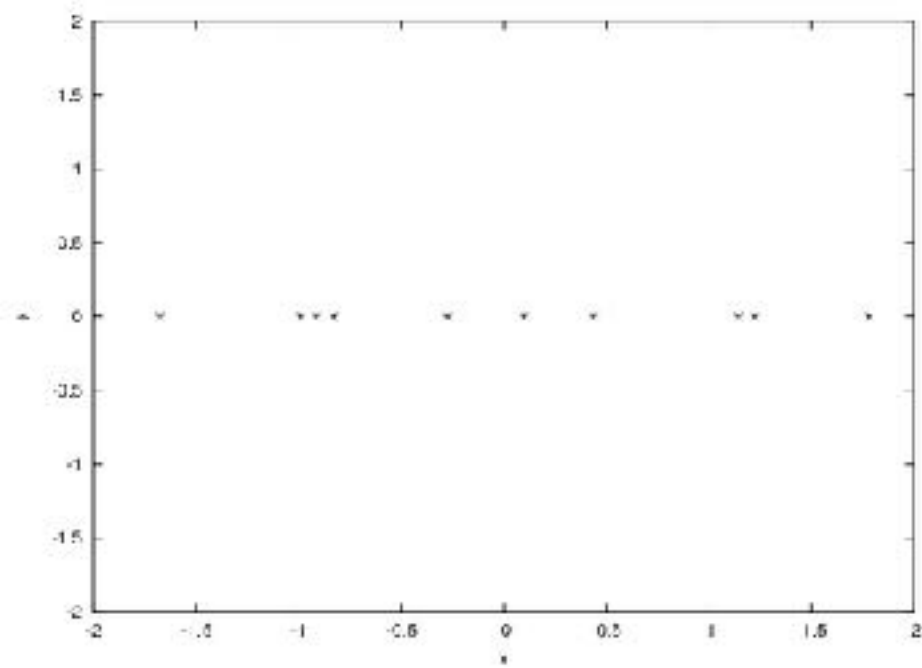


Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

Transformed Data (Single eigenvector)

x
.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22387056



Reconstructing the original data

We did:

$$\textit{TransformedData} = \textit{RowFeatureVector} \times \textit{RowDataAdjust}$$

so we can do

$$\textit{RowDataAdjust} = \textit{RowFeatureVector}^{-1} \times \textit{TransformedData}$$

$$= \textit{RowFeatureVector}^T \times \textit{TransformedData}$$

and

$$\textit{RowDataOriginal} = \textit{RowDataAdjust} + \textit{OriginalMean}$$

