Machine Learning Lecture 9

Dimensionality Reduction

Qian Ma

Sun Yat-sen University

Spring Semester, 2020

Acknowledgement

- A large part of slides in this lecture are originally from
  - Prof. Andrew Ng (Stanford University)
  - Prof. Shuai Li (Shanghai Jiao Tong University)

Prof. Andrew Ng
Stanford University

Prof. Shuai Li
Shanghai Jiao Tong University

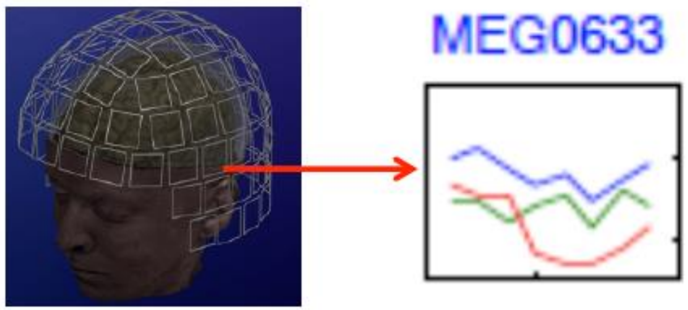
Motivation

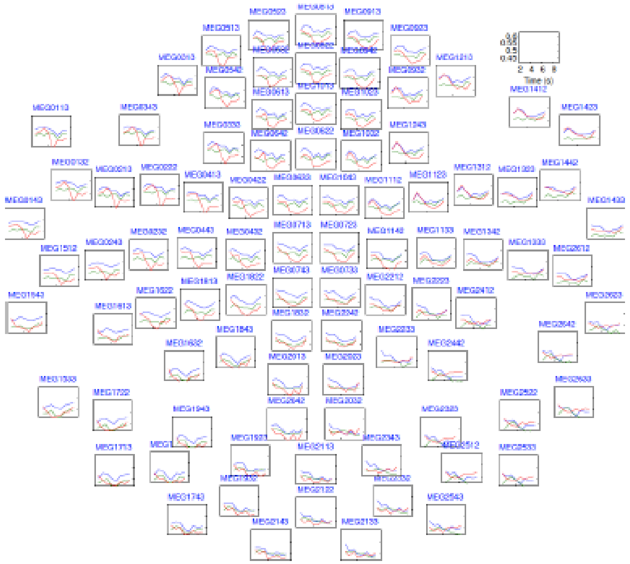
- Suppose we want to predict the health condition of some students, and the features for the students includes:
  - Weight in kilogram
  - Height in inch
  - Height in cm
  - Hours of sports per day
  - Favorite color
  - Scores in math

- Some features are irrelevant, e.g. favorite color and scores in math
- Some features are redundant, e.g. height in inch and cm

High dimensional data

- In the era of big data, the dimensionality increases dramatically
  - E.g. there are many features for the electroencephalogram data





- It becomes very important to reduce the dimensionality, or select the most important features, or find the most representative features

Principal Components Analysis

PCA

Principal components analysis (PCA)

- Principal components analysis (PCA) is a technique that can be used to simplify a dataset
- It is usually a linear transformation that chooses a new coordinate system for the data set such that
  - greatest variance by any projection of the dataset comes to lie on the first axis (then called the first principal component)
  - the second greatest variance on the second axis, and so on
- PCA can be used for reducing dimensionality by eliminating the later principal components

Example

- Consider the following 3D points

1	2	4	3	5	6
2	4	8	6	10	12
3	6	12	9	15	18

- If each component is stored in a byte, we need $18 = 3 \times 6$ bytes

Example (cont.)

- Looking closer, we can see that all the points are related geometrically
  - they are all in the same direction, scaled by a factor:

$$\begin{pmatrix}1\\2\\3\end{pmatrix} = 1 \times \begin{pmatrix}1\\2\\3\end{pmatrix} \qquad \begin{pmatrix}4\\8\\12\end{pmatrix} = 4 \times \begin{pmatrix}1\\2\\3\end{pmatrix} \qquad \begin{pmatrix}5\\10\\15\end{pmatrix} = 5 \times \begin{pmatrix}1\\2\\3\end{pmatrix}$$

$$\begin{pmatrix}2\\4\\6\end{pmatrix} = 2 \times \begin{pmatrix}1\\2\\3\end{pmatrix} \qquad \begin{pmatrix}3\\6\\9\end{pmatrix} = 3 \times \begin{pmatrix}1\\2\\3\end{pmatrix} \qquad \begin{pmatrix}6\\12\\18\end{pmatrix} = 6 \times \begin{pmatrix}1\\2\\3\end{pmatrix}$$

Example (cont.)

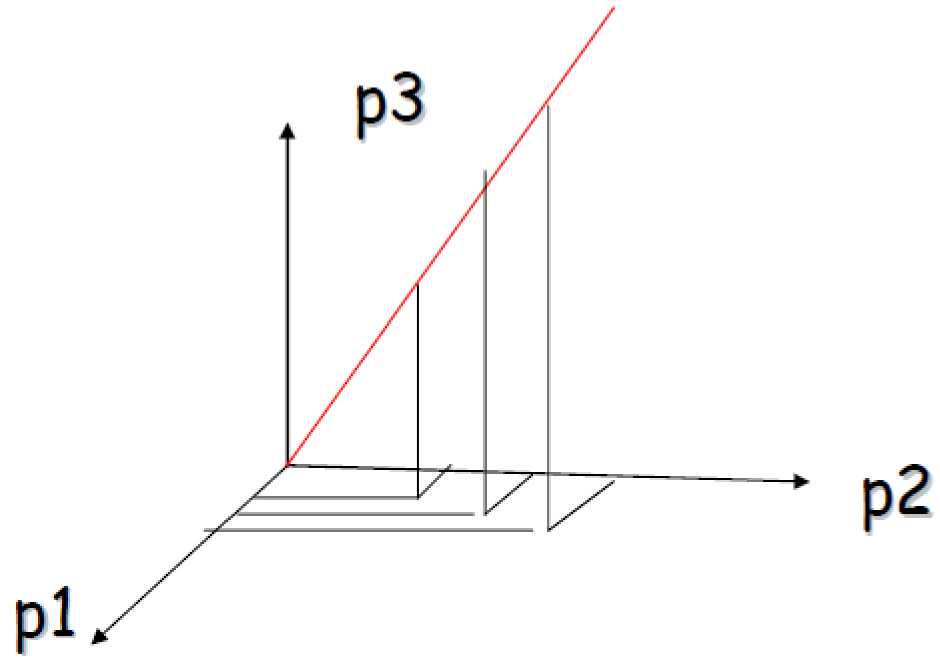
$$\begin{bmatrix}1\\2\\3\end{bmatrix} = 1 \times \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}4\\8\\12\end{bmatrix} = 4 \times \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}5\\10\\15\end{bmatrix} = 5 \times \begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$\begin{bmatrix}2\\4\\6\end{bmatrix} = 2 \times \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}3\\6\\9\end{bmatrix} = 3 \times \begin{bmatrix}1\\2\\3\end{bmatrix} \qquad \begin{bmatrix}6\\12\\18\end{bmatrix} = 6 \times \begin{bmatrix}1\\2\\3\end{bmatrix}$$

- They can be stored using only 9 bytes (50% savings!):
  - Store one direction (3 bytes) + the multiplying constants (6 bytes)

Geometrical interpretation

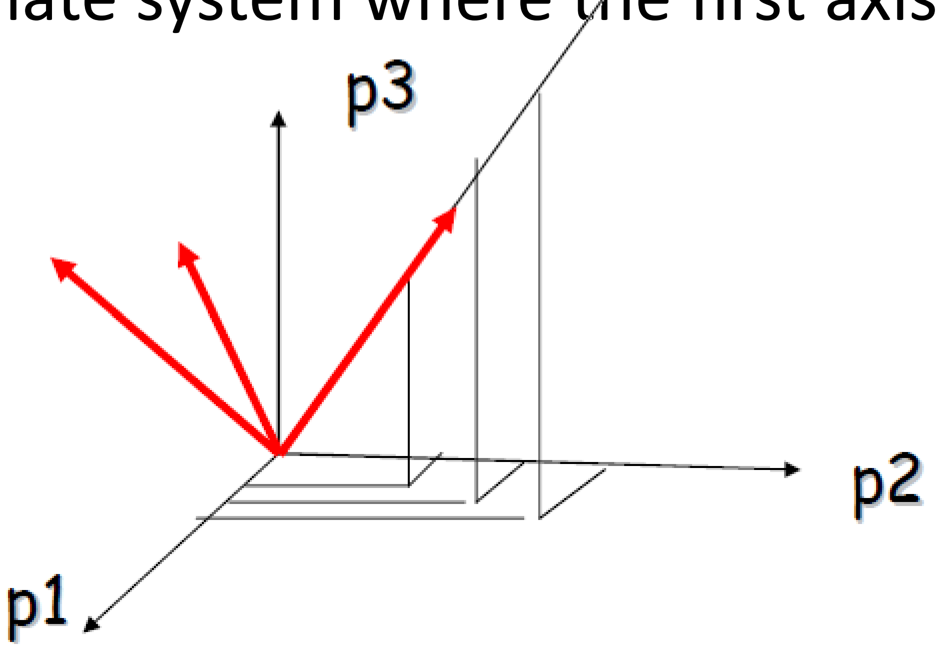
- View points in 3D space



- In this example, all the points happen to lie on one line
  - a 1D subspace of the original 3D space

Geometrical interpretation

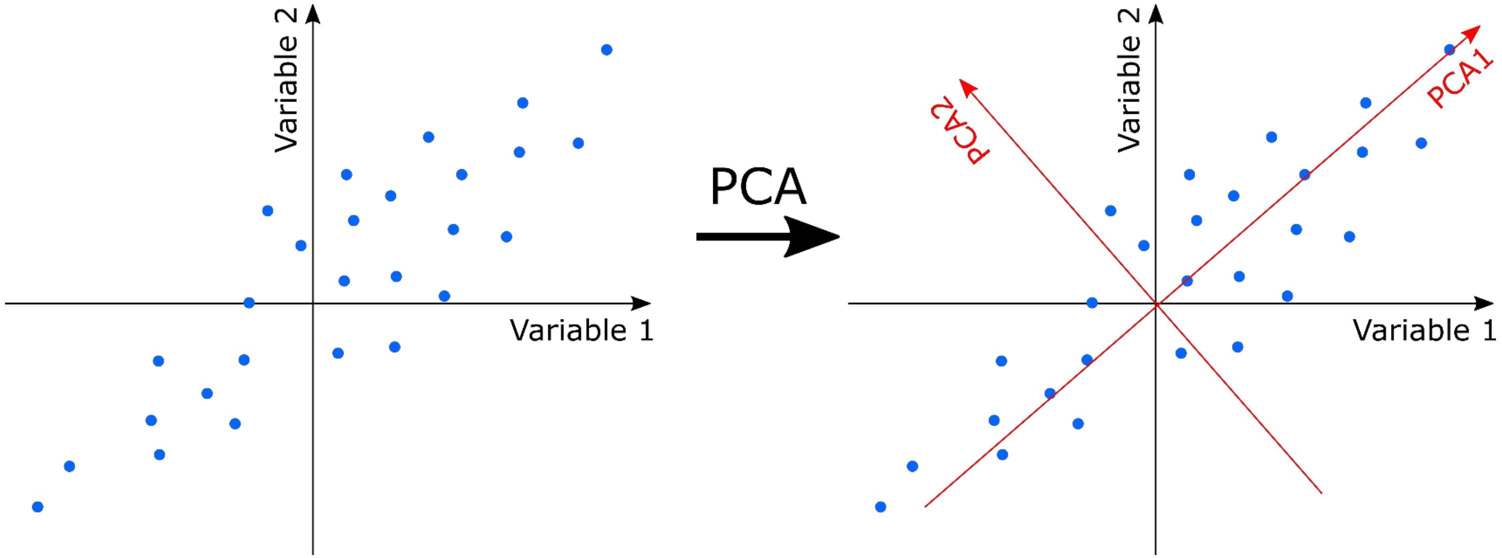
- Consider a new coordinate system where the first axis is along the direction of the line



- In the new coordinate system, every point has only one non-zero coordinate
  - we only need to store the direction of the line (a 3 bytes point) and the nonzero coordinates for each point (6 bytes)

Back to PCA

- Given a set of points, how can we know if they can be compressed similarly to the previous example?
  - We can look into the correlation between the points by the tool of PCA



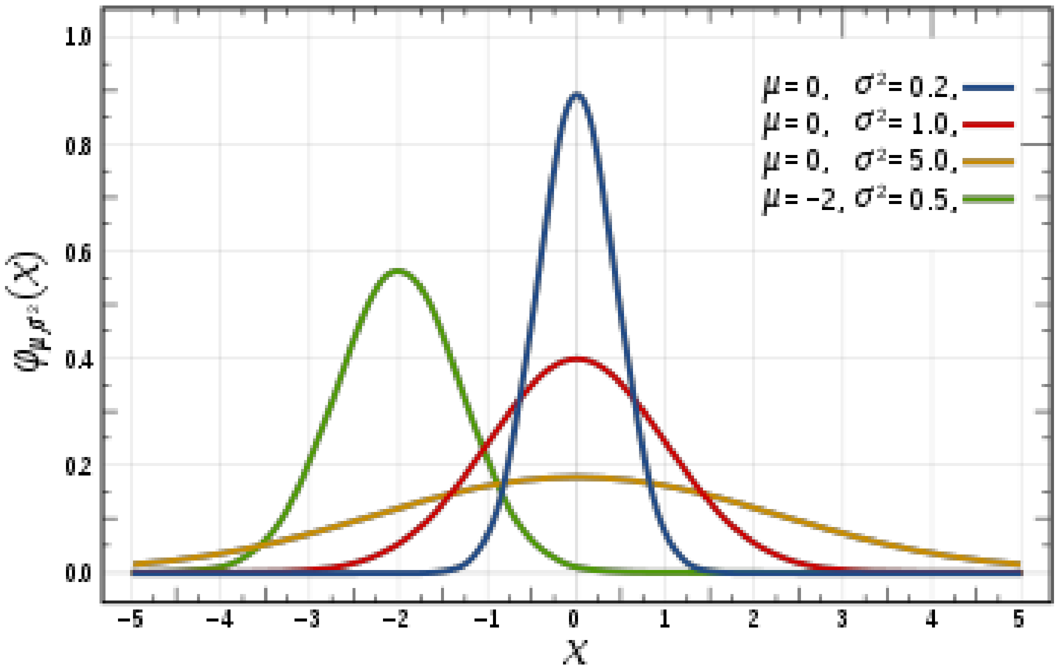
From example to theory

- In previous example, PCA rebuilds the coordination system for the data by selecting
  - the direction with largest variance as the first new base direction
  - the direction with the second largest variance as the second new base direction
  - and so on

- Then how can we find the direction with largest variance?
  - By the eigenvector for the covariance matrix of the data

Review – Variance

- Variance is the expectation of the squared deviation of a random variable from its mean
  - Informally, it measures how far a set of (random) numbers are spread out from their average value



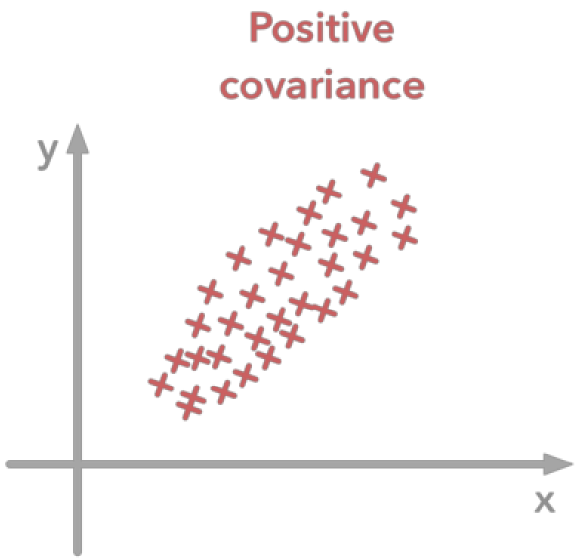
Review – Covariance

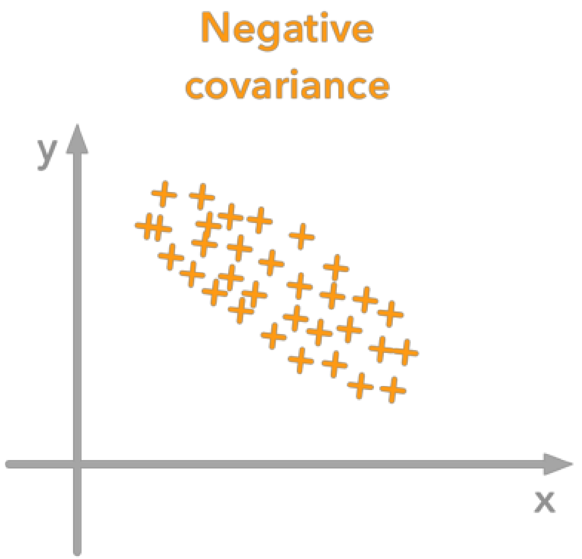
- Covariance is a measure of the joint variability of two random variables
  - If the greater values of one variable mainly correspond with the greater values of the other variable, and the same holds for the lesser values, (i.e., the variables tend to show similar behavior), the covariance is positive
    - E.g. as the number of hours studied increases, the marks in that subject increase
  - In the opposite case, when the greater values of one variable mainly correspond to the lesser values of the other, (i.e., the variables tend to show opposite behavior), the covariance is negative
  - The sign of the covariance therefore shows the tendency in the linear relationship between the variables
  - The magnitude of the covariance is not easy to interpret because it is not normalized and hence depends on the magnitudes of the variables. The normalized version of the covariance, the correlation coefficient, however, shows by its magnitude the strength of the linear relation

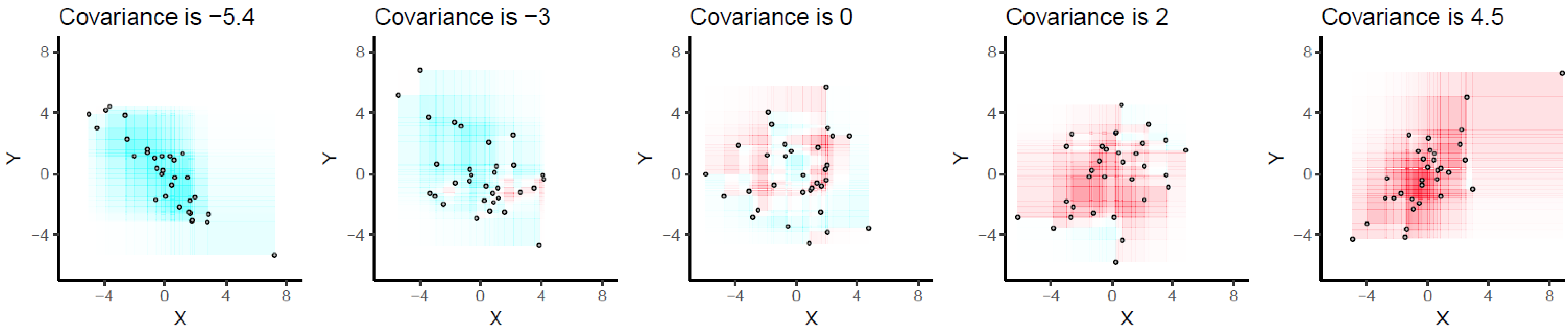
Review – Covariance (cont.)

- Sample covariance

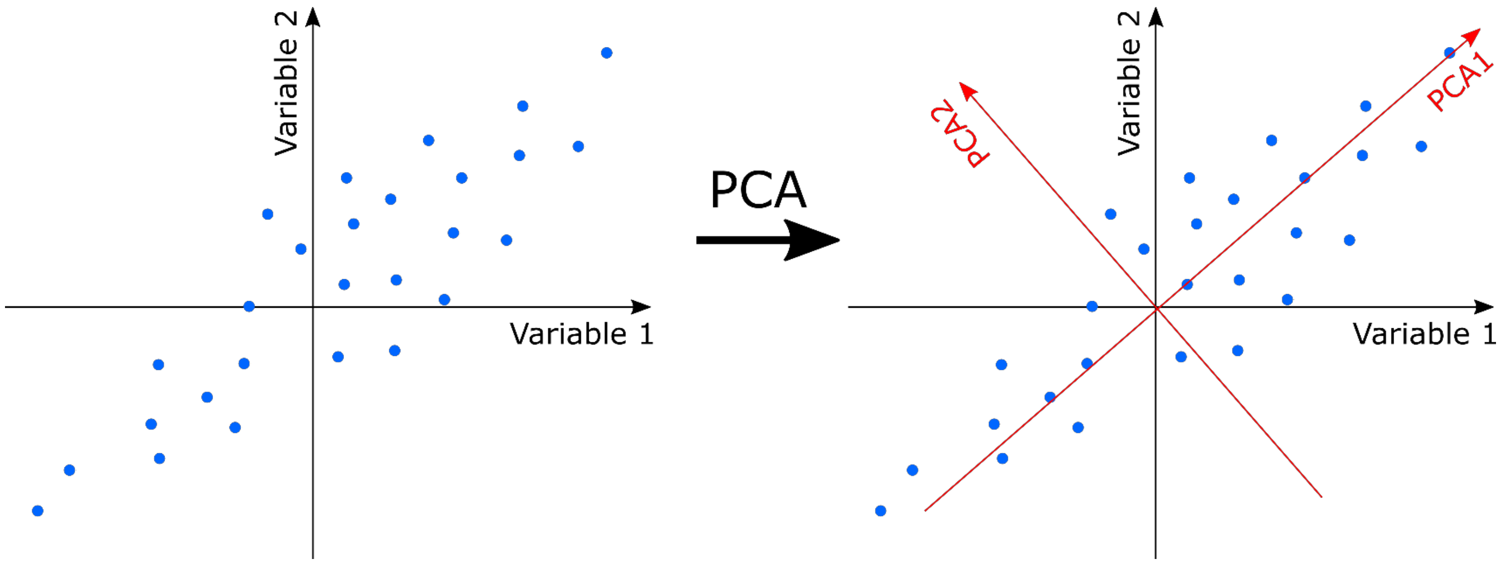
$$\text{covariance}(X,Y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$







PCA



- PCA tries to identify the subspace in which the data approximately lies in
- PCA uses an orthogonal transformation on the coordinate system to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components
  - The number of principal components is less than or equal to $\min\{d, N\}$

Covariance matrix

- Suppose there are 3 dimensions, denoted as X, Y, Z. The covariance matrix is

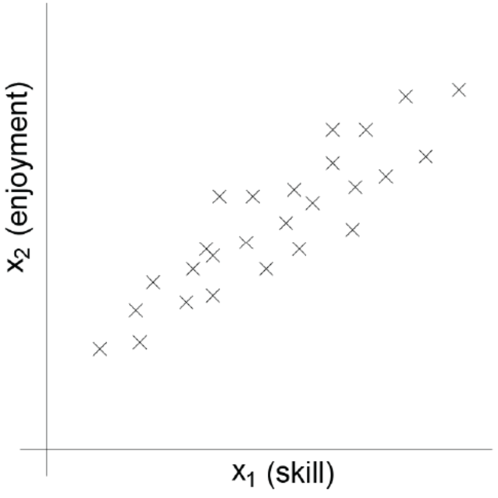
$$COV = \begin{bmatrix} COV(X,X) & COV(X,Y) & COV(X,Z) \\ COV(Y,X) & COV(Y,Y) & COV(Y,Z) \\ COV(Z,X) & COV(Z,Y) & COV(Z,Z) \end{bmatrix}$$

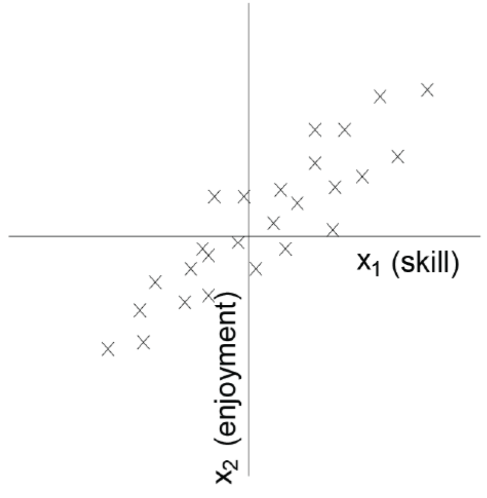
- Note the diagonal is the covariance of each dimension with respect to itself, which is just the variance of each random variable
- Also $COV(X,Y) = COV(Y,X)$
  - hence matrix is symmetric about the diagonal
- d-dimensional data will result in a $d \times d$ covariance matrix

Covariance in the covariance matrix

- Diagonal, or the variance, measures the deviation from the mean for data points in one dimension
- Covariance measures how one dimension random variable varies w.r.t. another, or if there is some linear relationship among them

Data processing





- Given the dataset $D = \left\{x^{(i)}\right\}_{i=1}^{N}$
- Let $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x^{(i)}$

$$X = \begin{bmatrix} \left(x^{(1)} - \bar{x}\right)^{\top} \\ \left(x^{(2)} - \bar{x}\right)^{\top} \\ \vdots \\ \left(x^{(N)} - \bar{x}\right)^{\top} \end{bmatrix} \in \mathbb{R}^{N \times d}$$

  - Move the center of the data set to 0

Data processing (cont.)

- $Q = X^\top X = \begin{bmatrix} x^{(1)} - \bar{x} & x^{(2)} - \bar{x} & \cdots & x^{(N)} - \bar{x} \end{bmatrix} \begin{bmatrix} \left(x^{(1)} - \bar{x}\right)^\top \\ \left(x^{(2)} - \bar{x}\right)^\top \\ \vdots \\ \left(x^{(N)} - \bar{x}\right)^\top \end{bmatrix}$
  - Q is square with d dimension
  - Q is symmetric
  - Q is the covariance matrix [aka scatter matrix]
  - Q can be very large (in vision, d is often the number of pixels in an image!)
    - For a 256×256 image, $d = 65536$!!
    - Don't want to explicitly compute Q

PCA

- By finding the eigenvalues and eigenvectors of the covariance matrix, we find that the eigenvectors with the largest eigenvalues correspond to the dimensions that have the strongest variation in the dataset
- This is the principal component
- Application:
  - face recognition, image compression
  - finding patterns in data of high dimension

Eigenvector of Matrix **A**

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

Eigenvalue of Matrix **A**

PCA theorem

- Theorem:
- Each $x^{(i)}$ can be written as: $x^{(i)} = \bar{x} + \sum_{j=1}^{d} g_{ij} e_j$
  where e_j are the d eigenvectors of Q with non-zero eigenvalues

- **Notes:**
  1. The eigenvectors $e_1 e_2 \cdots e_d$ span an **eigenspace**
  2. $e_1 e_2 \cdots e_d$ are $d \times 1$ orthonormal vectors (directions in d-Dimensional space)
  3. The scalars g_{ij} are the coordinates of $x^{(i)}$ in the space

$$g_{ij} = \langle x^{(i)} - \bar{x}, e_j \rangle$$

Using PCA to compress data

- Expressing x in terms of $e_1 e_2 \cdots e_d$ doesn't change the size of the data
- However, if the points are highly correlated, many of the new coordinates of x will become zero or close to zero
- Sort the eigenvectors e_i according to their eigenvalue

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$$

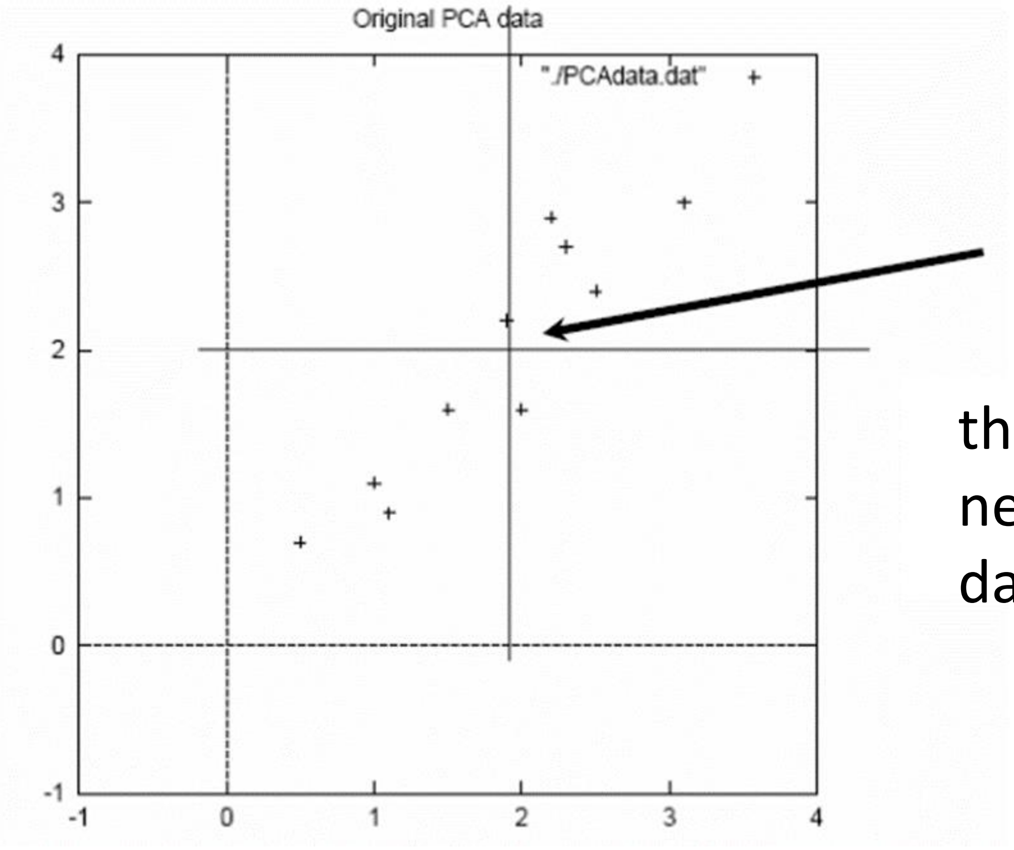
- Assume $\lambda_j \approx 0$ if $j > k$. Then

$$x^{(i)} \approx \bar{x} + \sum_{j=1}^{k} g_{ij} e_j$$

Example – STEP 1

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

this becomes the new origin of the data from now on

http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf

Example – STEP 2

- Calculate the covariance matrix

$$\mathrm{Cov} = \begin{bmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{bmatrix}$$

- since $\mathrm{cov}(X, Y)$ is positive, it is expect that x and y increase together

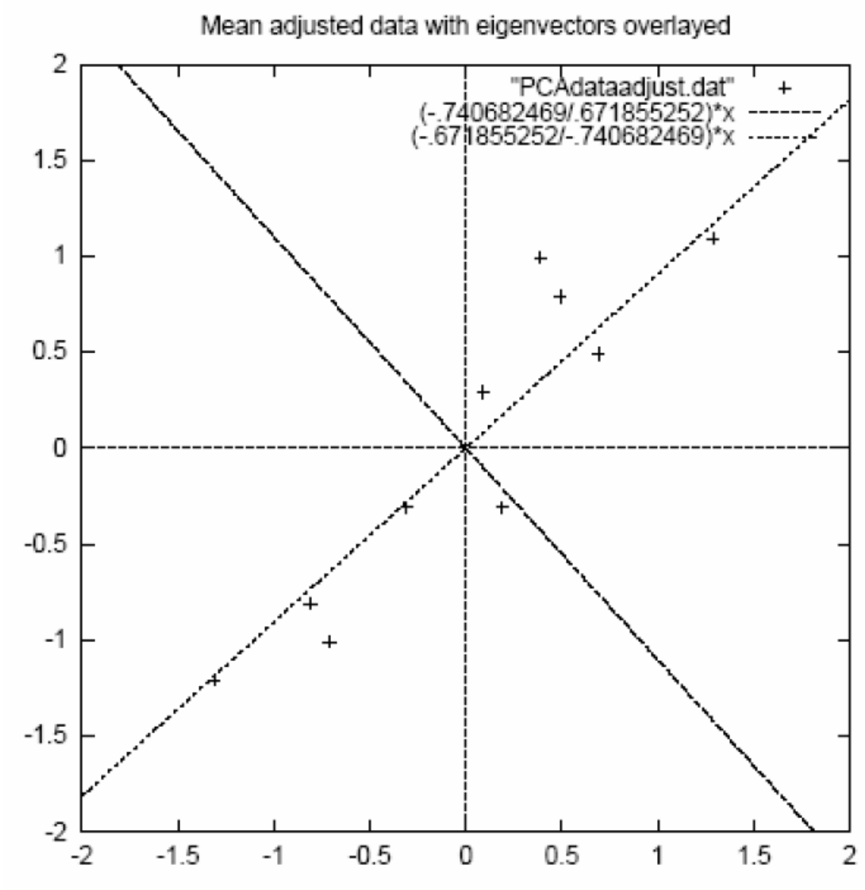
Example – STEP 3

- Calculate the eigenvectors and eigenvalues of the covariance matrix

- $\text{eigenvalues} = \begin{bmatrix} 0.0490833989 \\ 1.28402771 \end{bmatrix}$

- $\text{eigenvectors} = \begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$

Example – STEP 3 (cont.)



- Eigenvectors are plotted as diagonal dotted lines on the plot
- Note they are perpendicular to each other
- Note one of the eigenvectors goes through the middle of the points, like drawing a line of best fit
- The second eigenvector gives us the other, less important, pattern in the data, that all the points follow the main line, but are off to the side of the main line by some amount

Example – STEP 4

- Feature vector $= [e_1 \quad e_2 \quad \cdots \quad e_d]$

- We can either form a feature vector with both of the eigenvectors:

$$\begin{bmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{bmatrix}$$

- or, we can choose to delete the smaller, less significant component:

$$\begin{bmatrix} -0.677873399 \\ -0.735178656 \end{bmatrix}$$

Example – STEP 5

$$\text{FinalData}_{N\times d} = \begin{bmatrix} g\left(x^{(1)}\right)^\top \\ \vdots \\ g\left(x^{(N)}\right)^\top \end{bmatrix}_{N\times d} \begin{bmatrix} e_1^\top \\ \vdots \\ e_d^\top \end{bmatrix}_{d\times d}$$

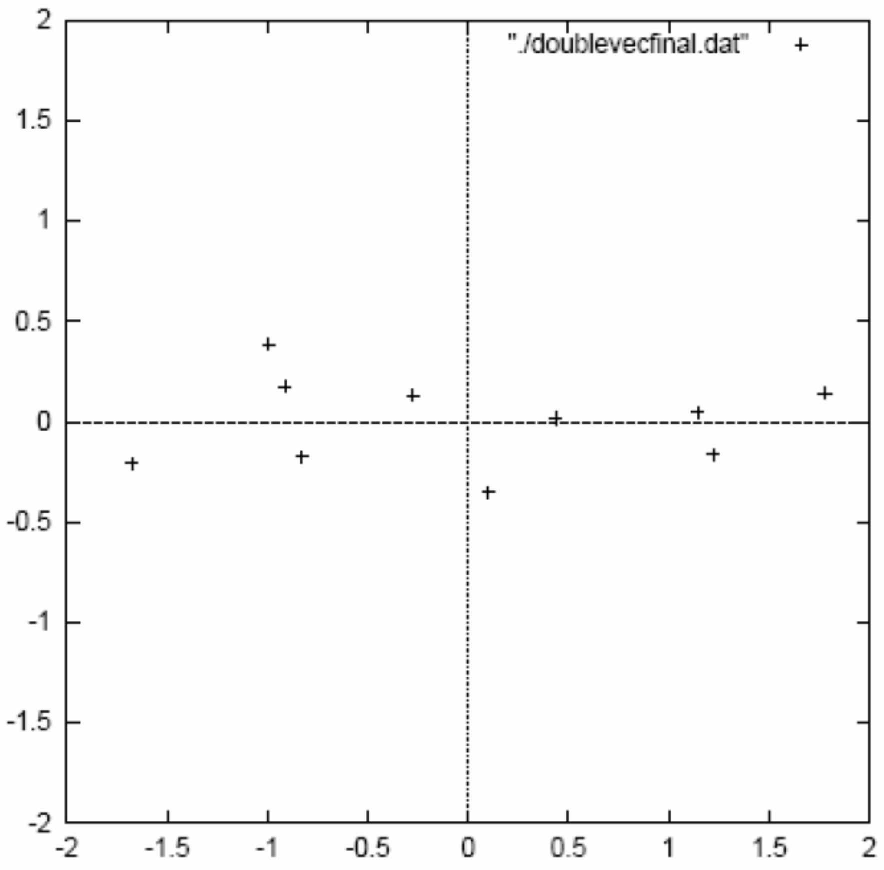
- Deriving new data coordinates

  FinalData = RowZeroMeanData x RowFeatureVector

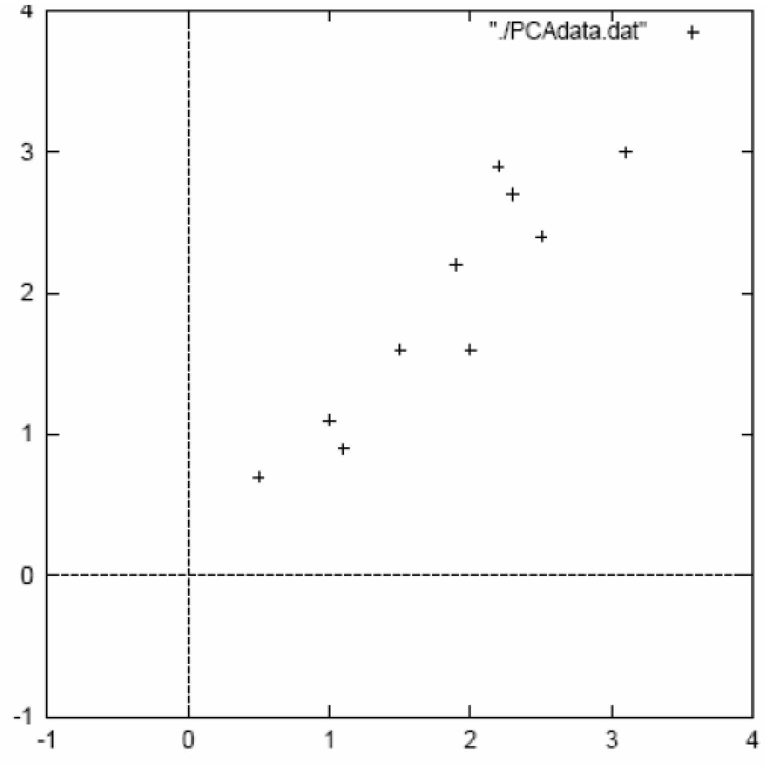
- RowZeroMeanData is the mean-adjusted data, i.e. the data items are in each row, with each column representing a separate dimension
- RowFeatureVector is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top
- Note: We rotate the coordinate axes so high-variance axis comes first

Example – STEP 5 (cont.)

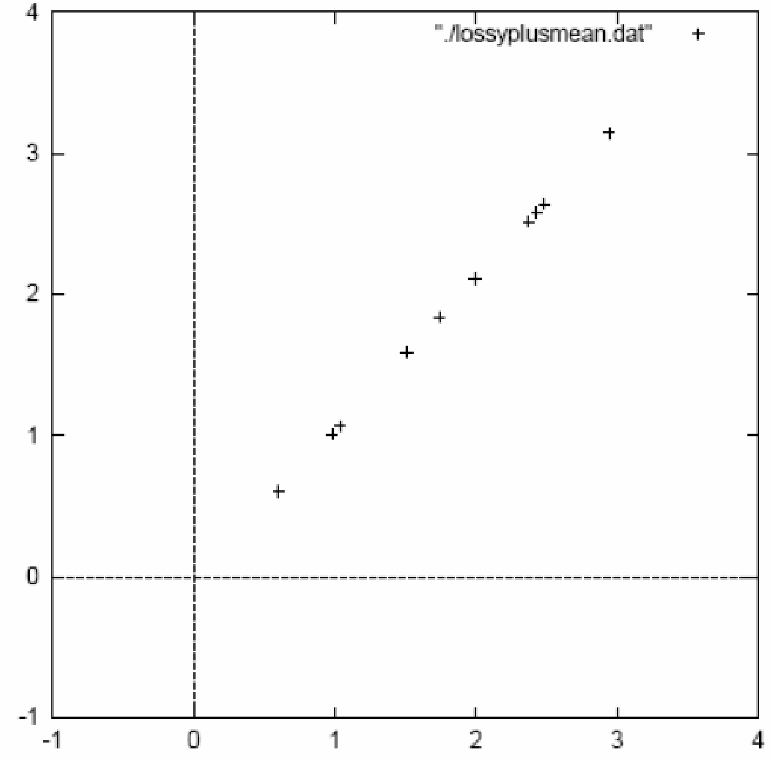
- The plot of the PCA results using both the two eigenvector



Example – Final approximation



2D point cloud



Approximation using one eigenvector basis

Example – Final approximation

$$\text{FinalData}_{N\times d} = \begin{bmatrix} g\left(x^{(1)}\right)^{\top} \\ \vdots \\ g\left(x^{(N)}\right)^{\top} \end{bmatrix}_{N\times d} \begin{bmatrix} e_1^{\top} \\ \vdots \\ e_d^{\top} \end{bmatrix}_{d\times d}$$

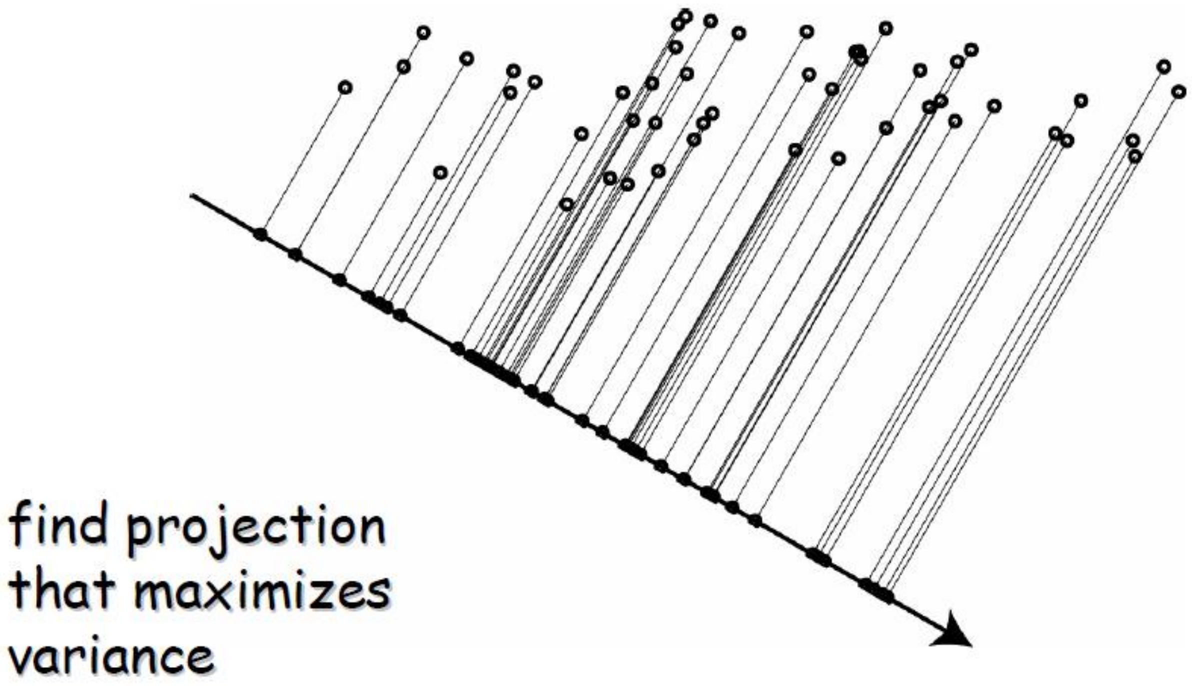
$$\approx \begin{bmatrix} g\left(x^{(1)}\right)_1 & \cdots & g\left(x^{(1)}\right)_k & \cancel{\cdots} & \cancel{g\left(x^{(1)}\right)_d} \\ & & \vdots & & \\ g\left(x^{(N)}\right)_1 & \cdots & g\left(x^{(N)}\right)_k & \cancel{\cdots} & \cancel{g\left(x^{(N)}\right)_d} \end{bmatrix}_{N\times d} \begin{bmatrix} e_1^{\top} \\ \vdots \\ e_k^{\top} \\ \vdots \\ e_d^{\top} \end{bmatrix}_{d\times d}$$

Zero!!

Revisit the eigenvectors in PCA

- It is critical to notice that the *direction of maximum variance* in the input space happens to be same as the *principal eigenvector of the covariance matrix*

- Why?



Revisit the eigenvectors in PCA (cont.)

- The projection of each point x to a direction u (with $\|u\| = 1$) is
$$x^\top u$$
- The variance of the projection is
$$\sum_{i=1}^{N} \left(\left(x^{(i)} - \bar{x} \right)^\top u \right)^2 = u^\top Q u$$
which is maximized when u is the eigenvector with the largest eigenvalue
- $Q = \sum_{j=1}^{d} \lambda_j e_j {e_j}^\top = E \Lambda E^\top$ with $\Lambda = \begin{bmatrix} \lambda_1 & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \lambda_d \end{bmatrix}$

Review – Total/Explained variance

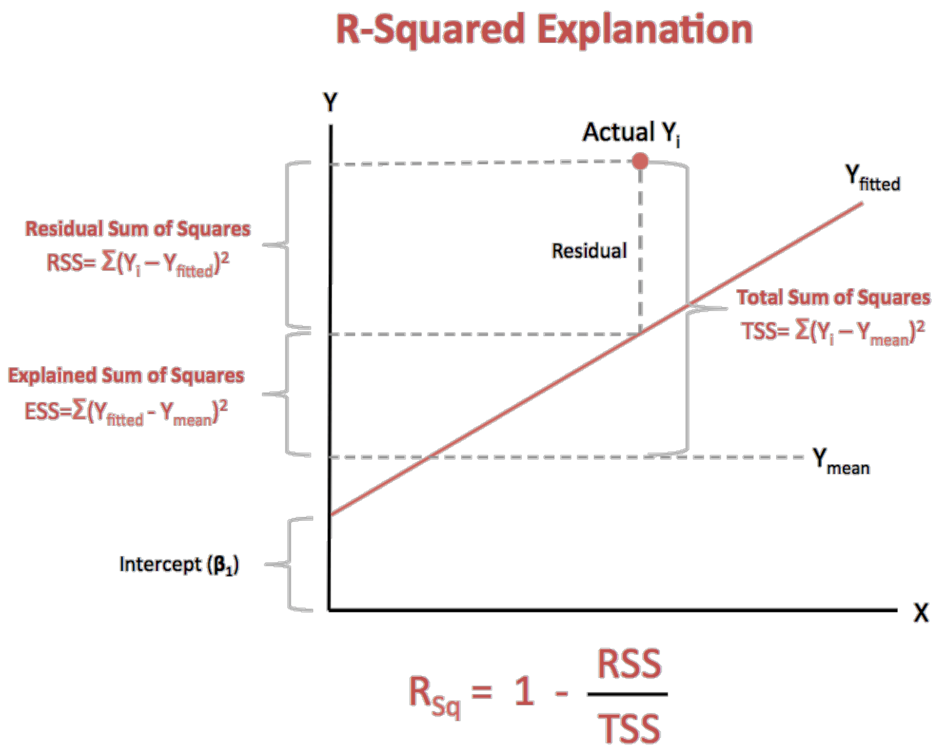
- $R^2 = \frac{explained\ variance}{total\ variance}$

- Total variance: $SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$

- Explained variance: $SS_{\text{reg}} = \sum_i (f_i - \bar{y})^2$

- Or, it can be computed as:

$R^2 \equiv 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$ where $SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$



Total variance and PCA

- Note that $I = e_1 e_1^\top + \cdots + e_d e_d^\top$
- Total variance is
- $\sum_{i=1}^{N} \left(x^{(i)} - \bar{x}\right)^\top \left(x^{(i)} - \bar{x}\right)$
- $= \sum_{i=1}^{N} \left(x^{(i)} - \bar{x}\right)^\top \left(e_1 e_1^\top + \cdots + e_d e_d^\top\right)\left(x^{(i)} - \bar{x}\right)$
- $= \sum_{j=1}^{d} e_j^\top Q\, e_j = \lambda_1 + \cdots + \lambda_d$

Total variance and PCA (cont.)

- Approximation of each $x^{(i)} - \bar{x} \approx \sum_{j=1}^{k} g_{ij} e_j =: \tilde{x}^{(i)} - \bar{x}$
- Then the explained variance is
- $\sum_{i=1}^{N} \left(\tilde{x}^{(i)} - \bar{x}\right)^{\top} \left(\tilde{x}^{(i)} - \bar{x}\right)$
- $= \sum_{i=1}^{N} \left(\tilde{x}^{(i)} - \bar{x}\right)^{\top} \left(e_1 e_1^{\top} + \cdots + e_d e_d^{\top}\right)\left(\tilde{x}^{(i)} - \bar{x}\right)$
- $= \sum_{j=1}^{d} e_j^{\top} \tilde{Q} e_j = \lambda_1 + \cdots + \lambda_k$
- where $\tilde{Q} = \sum_{i=1}^{N} \left(\tilde{x}^{(i)} - \bar{x}\right) \left(\tilde{x}^{(i)} - \bar{x}\right)^{\top} = E \tilde{\Lambda} E^{\top}$ with
- $\tilde{\Lambda} = \begin{bmatrix} \lambda_1 & & & & \\ & \ddots & & & \\ & & \lambda_k & & \\ & & & 0 & \\ & & & & 0 \end{bmatrix}$