Principal Component Analysis

Dr. Fayyaz Minhas

Department of Computer Science
University of Warwick
https://warwick.ac.uk/fac/sci/dcs/teaching/material/cs909/

Question?

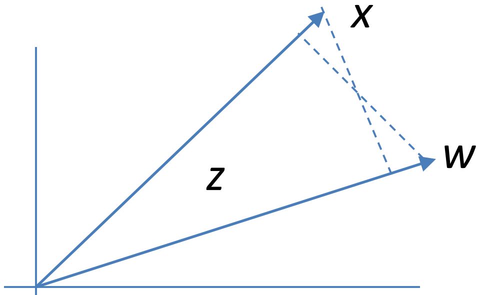
- Consider the vectors
  - $X_1=[1\ 2\ 1\ 4]^T$
  - $X_2=[2\ 4\ 2\ 4]^T$
  - $X_3=[0\ 0\ 0\ 4]^T$
  - $X_4=[3\ 6\ 3\ 4]^T$
  - $X_5=[4\ 8\ 4\ 4]^T$
- To store each vector, how many dimensions (or variables) do we need?

Motivation

- Having large number of related features is not informative
- Can we reduce the number of features?
  - Dimensionality Reduction

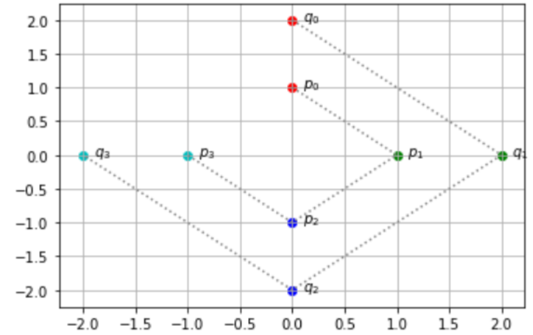
Basics

- Projections
  - Data can be projected onto a vector by taking its dot-product
    - The ith-component of a data point is the projection of the data onto the vector corresponding to the i-th axis
  - $z = w^T x$
    - Projection of x in the direction of w



- Transformation
  - Multiplication of a vector with a matrix can be viewed as a geometric transformation of the vector

$$\boldsymbol{T} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\boldsymbol{y} = \boldsymbol{T}\boldsymbol{x} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$



- Eigen Values and Vectors
  - Those points that are characteristic to a given matrix that undergo only a change in scale are called Eigen vectors $\boldsymbol{w} = \boldsymbol{T}\boldsymbol{v} = \lambda \boldsymbol{v}$
  - How to find them: $(\boldsymbol{T} - \lambda \boldsymbol{I})\boldsymbol{v} = 0$ implies $|\boldsymbol{T} - \lambda \boldsymbol{I}| = 0$

Example: Find Eigenvalues and Eigenvectors of a 2x2 Matrix

If

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$$

then the characteristic equation is

$$|\mathbf{A} - \lambda \cdot \mathbf{I}| = \left| \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right| = 0$$

$$\left| \begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix} \right| = \lambda^2 + 3\lambda + 2 = 0$$

and the two eigenvalues are

$$\lambda_1=-1,\ \lambda_2=-2$$

All that's left is to find the two eigenvectors. Let's find the eigenvector, $\mathbf{v}_1$, associated with the eigenvalue, $\lambda_1=-1$, first.

$$\mathbf{A} \cdot \mathbf{v}_1 = \lambda_1 \cdot \mathbf{v}_1$$

$$(\mathbf{A} - \lambda_1) \cdot \mathbf{v}_1 = 0$$

$$\begin{bmatrix} -\lambda_1 & 1 \\ -2 & -3-\lambda_1 \end{bmatrix} \cdot \mathbf{v}_1 = 0$$

$$\begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix} \cdot \mathbf{v}_1 = \begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix} \cdot \begin{bmatrix} v_{1,1} \\ v_{1,2} \end{bmatrix} = 0$$

so clearly from the top row of the equations we get

$$v_{1,1} + v_{1,2} = 0, \quad \text{so}$$
$$v_{1,1} = -v_{1,2}$$

Note that if we took the second row we would get

$$-2 \cdot v_{1,1} + -2 \cdot v_{1,2} = 0, \quad \text{so again}$$
$$v_{1,1} = -v_{1,2}$$

In either case we find that the first eigenvector is any 2 element column vector in which the two elements have equal magnitude and opposite sign.

$$\mathbf{v}_1 = k_1 \begin{bmatrix} +1 \\ -1 \end{bmatrix}$$

where k_1 is an arbitrary constant. Note that we didn't have to use +1 and -1, we could have used any two quantities of equal magnitude and opposite sign.

Going through the same procedure for the second eigenvalue:

$$\mathbf{A} \cdot \mathbf{v}_2 = \lambda_2 \cdot \mathbf{v}_2$$

$$(\mathbf{A} - \lambda_2) \cdot \mathbf{v}_2 = \begin{bmatrix} -\lambda_2 & 1 \\ -2 & -3-\lambda_2 \end{bmatrix} \cdot \mathbf{v}_2 = \begin{bmatrix} 2 & 1 \\ -2 & -1 \end{bmatrix} \cdot \begin{bmatrix} v_{2,1} \\ v_{2,2} \end{bmatrix} = 0 \quad \text{so}$$

$$2 \cdot v_{2,1} + 1 \cdot v_{2,2} = 0 \quad (\text{or from bottom line: } -2 \cdot v_{2,1} - 1 \cdot v_{2,2} = 0)$$

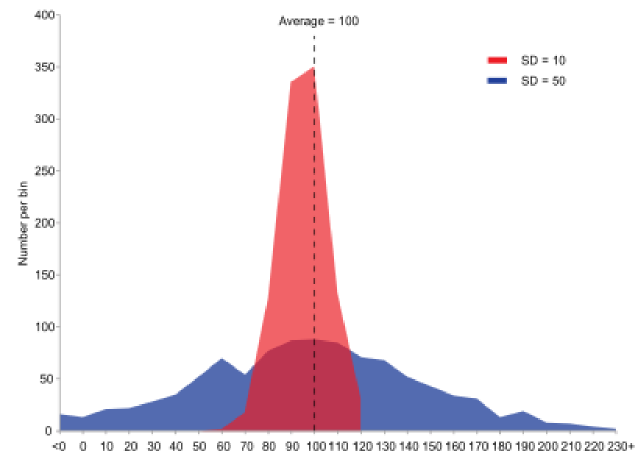
$$2 \cdot v_{2,1} = -v_{2,2}$$

$$\mathbf{v}_2 = k_2 \begin{bmatrix} +1 \\ -2 \end{bmatrix}$$

Again, the choice of +1 and -2 for the eigenvector was arbitrary; only their ratio is important. This is demonstrated in the MatLab code below.

Basics

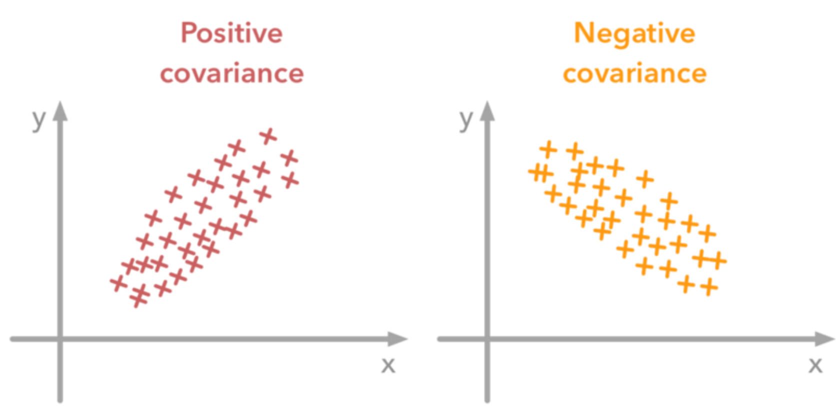
- Variance
  - Mean of the spread of a variable around its mean
  - $var(z) = \frac{1}{N}\sum_{i=1}^{N}(z_i - \mu_z)^2 = \frac{1}{N}(\boldsymbol{z} - \mu_z)^T(\boldsymbol{z} - \mu_z)$
    - $\boldsymbol{z}$ is an N-dimensional vector composed of the values of all data points in the sample
  - If mean is zero then $var(z) = \frac{1}{N}\boldsymbol{z}^T\boldsymbol{z} = \frac{1}{N}\|\boldsymbol{z}\|^2$
  - $var(z) = E[(z - \mu_z)^2]$



- Variance as an information measure
  - How is variance related to information content?

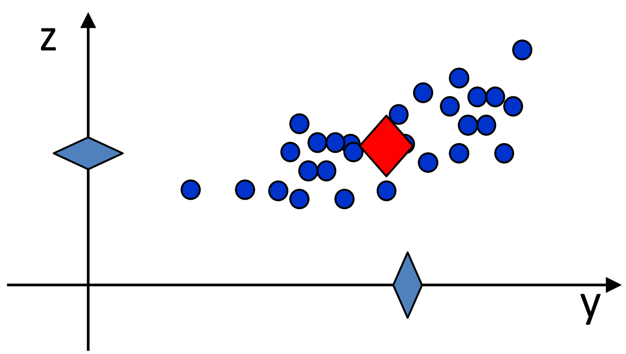
Covariance

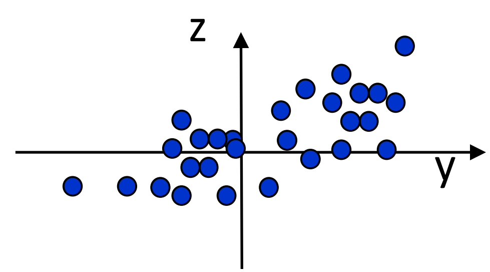
- Co-Variance
  - Given two random variables, to what extent are they linearly related to each other
  - $cov(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) = \frac{1}{N}(\boldsymbol{x} - \mu_x)^T(\boldsymbol{y} - \mu_y)$
  - Assume that the means are zero: $cov(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{N}\boldsymbol{x}^T\boldsymbol{y}$
    - Maximum when the vectors are co-linear or parallel
  - $cov(\boldsymbol{x}, \boldsymbol{y}) = E[(y - \mu_y)(x - \mu_x)]$
  - Thus, $var(z) = cov(z, z)$



Basics

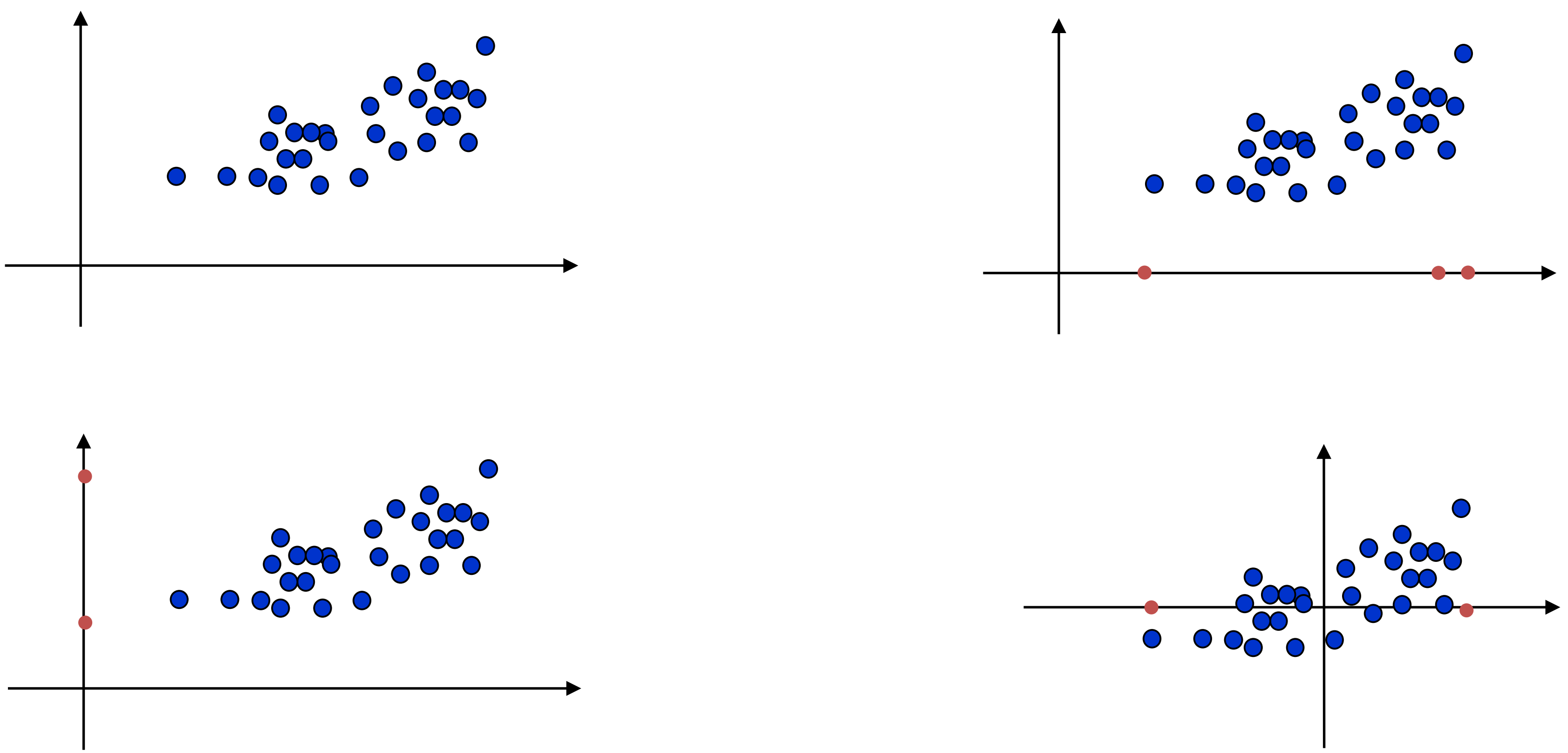
- Why are we interested in covariance
  - If two variables co-vary then they are redundant or one can be linearly deduced from another
  - Covariance matrix: matrix of all pairwise covariances of all variables
    - $\boldsymbol{C} = \begin{bmatrix} cov(y,y) & cov(z,y) \\ cov(y,z) & cov(z,z) \end{bmatrix}$





Data Dimensionality Reduction

- How can we reduce dimensions?

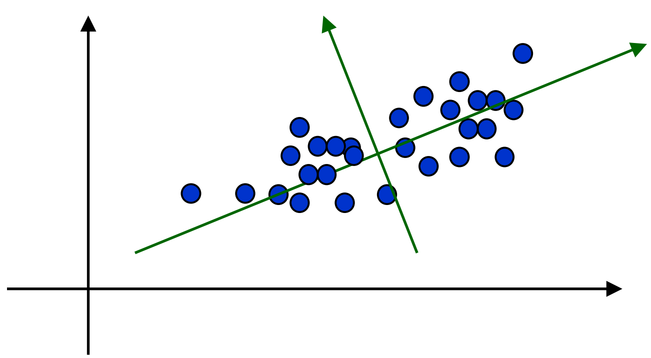


Dimensionality Reduction as Projections

- Projections can be used for reducing dimensions
  - However, projecting data onto a vector loses information
  - We want to reduce the amount of information loss
  - Solution: Find and project along a direction along which information loss is minimum

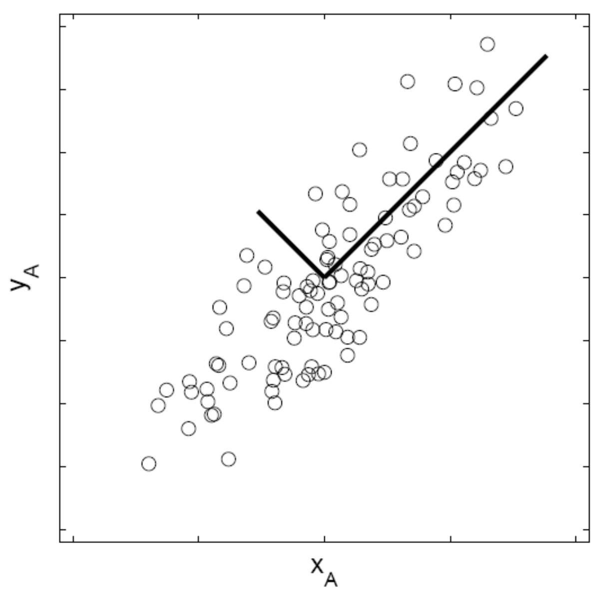
So what is PCA?

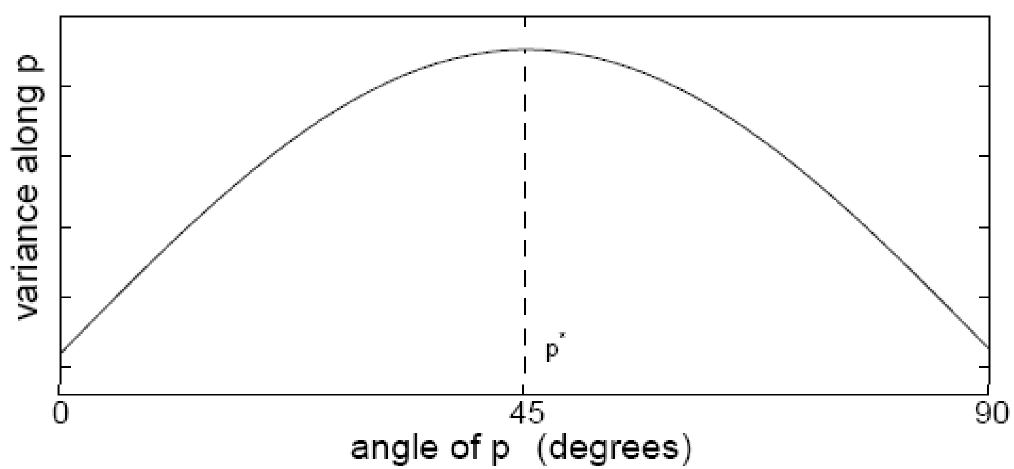
- A method for transforming the data
  - Projecting the data onto a vector such that the variance of the projected data is maximum
    - Because variance is proportional to information content



Principal Component Analysis

- Finding the directions w on which the data is to be projected through variance maximization makes the projected data most spread out so that the difference between points becomes most apparent





Principal Component Analysis

- Find a low-dimensional space such that when x is projected there, information loss is minimized.
- The projection of x on the direction of w is: $z = w^T x$
- Find w such that Var(z) is maximized

$$\begin{aligned}
\mathrm{Var}(z) = \mathrm{Var}(w^T x) &= E[(w^T x - w^T \mu)^2] \\
&= E[(w^T x - w^T \mu)(w^T x - w^T \mu)] \\
&= E[w^T (x - \mu)(x - \mu)^T w] \\
&= w^{\mathrm{T}} E[(x - \mu)(x - \mu)^T] w = w^T \mathrm{C}\, w
\end{aligned}$$

where $\mathrm{Cov}(x) = E[(x - \mu)(x - \mu)^T] = \mathrm{C}$

Principal Component Analysis

- Maximize Var(z_1) subject to $||w||$=1

$$\max_{\mathbf{w}_1} \ \mathbf{w}_1^T \boldsymbol{C} \mathbf{w}_1 - \alpha(\mathbf{w}_1^T \mathbf{w}_1 - 1)$$

$Cw_1 = \alpha w_1$

Differentiating w.r.t $\mathbf{w}_1$

  - w_1 is an eigenvector of $\sum$
  - Choose the one with the largest eigenvalue for Var(z_1) to be max
- Second principal component: Max Var(z_2), s.t., $||w_2||$=1 and orthogonal to w_1

$$\max_{\mathbf{w}_2} \mathbf{w}_2^T \boldsymbol{C} \mathbf{w}_2 - \alpha(\mathbf{w}_2^T \mathbf{w}_2 - 1) - \beta(\mathbf{w}_2^T \mathbf{w}_1 - 0)$$

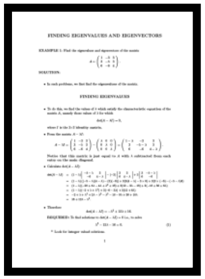
$Cw_2 = \alpha \ w_2$

  - w_2 is another eigenvector of $\sum$

$$\begin{aligned}
&\Rightarrow \quad 2\Sigma\mathbf{w}_2 - 2\alpha\mathbf{w}_2 - \beta\mathbf{w}_1 = 0 \\
&\Rightarrow \quad 2\mathbf{w}_1^T\Sigma\mathbf{w}_2 - 2\alpha\mathbf{w}_1^T\mathbf{w}_2 - \beta\mathbf{w}_1^T\mathbf{w}_1 = 0 \\
&\Rightarrow \quad 2\mathbf{w}_2^T\Sigma\mathbf{w}_1 - 2\alpha(0) - \beta(1) = 0 \\
&\Rightarrow \quad 2\lambda_1\mathbf{w}_2^T\mathbf{w}_1 - \beta = 0 \qquad \Rightarrow \beta = 0
\end{aligned}$$

- And so on. The Eigen values are sorted in decreasing order and the eigen vectors with positive eigen values are kept

Supporting Material



FINDING EIGENVALUES AND EIGENVECTORS

Covariance

D.Click after exiting slide show for a tutorial on
Eigen Values & Eigen Vectors

D.Click after exiting slide show for a tutorial on
Variance & Covariance

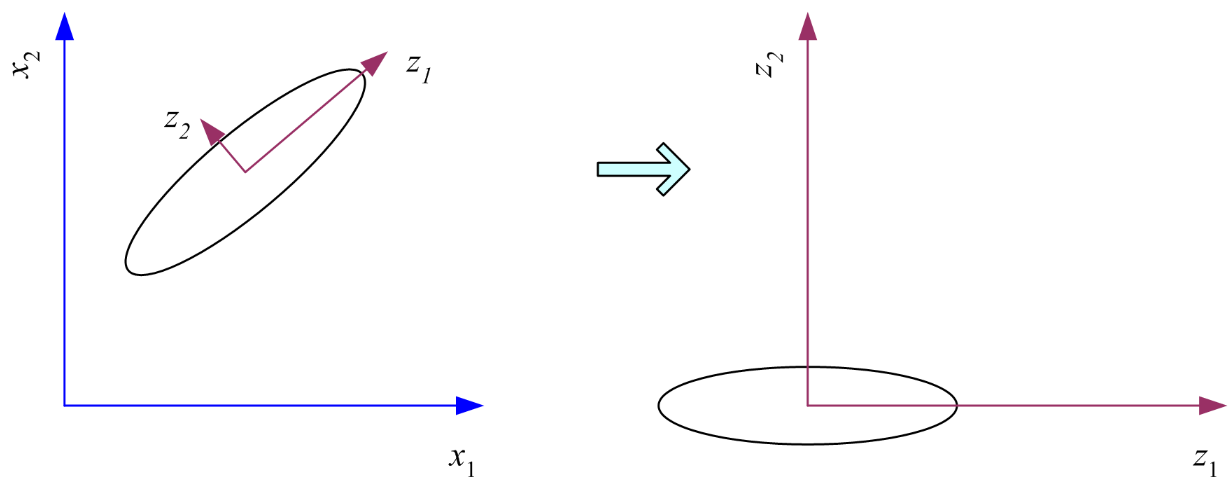
Effects of PCA

- In PCA we project the given vector x using

  $z = W^T(x - m)$

  where the columns of W are the eigenvectors of Σ, and m is sample mean

- PCA Centers the data at the origin and rotates the axes to match directions of maximal variance

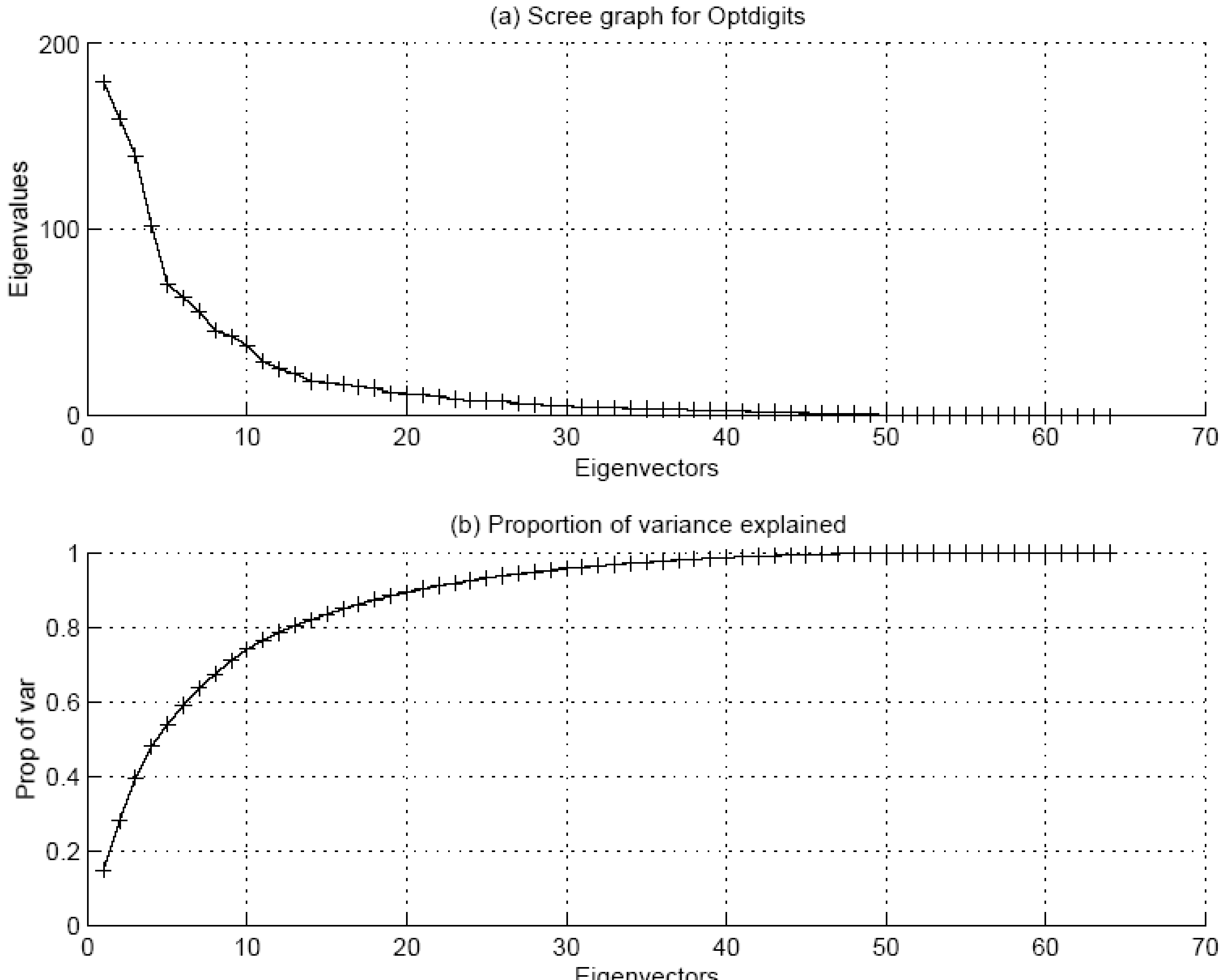


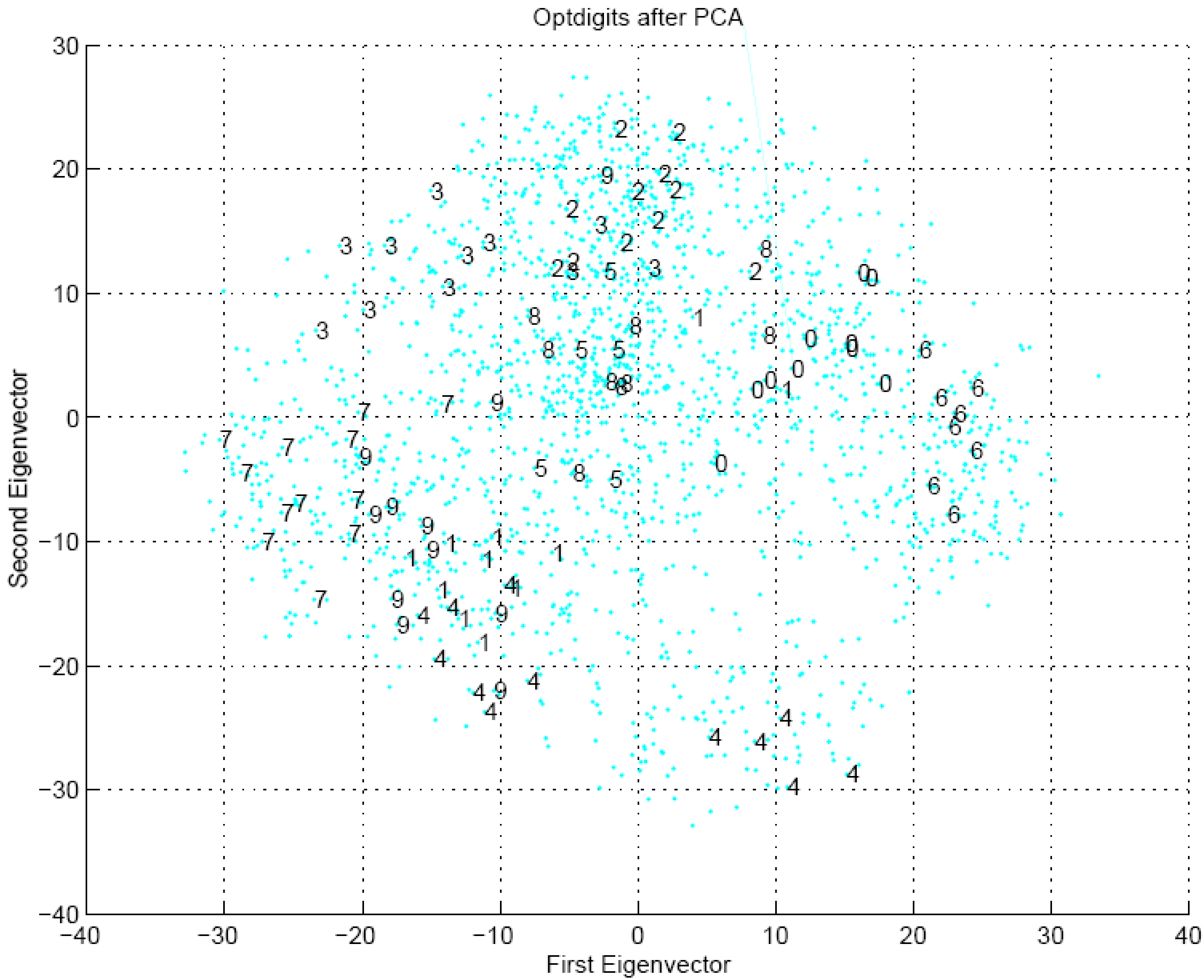
PCA for dimensionality reduction

- Eigen values of the covariance matrix with small magnitudes have small contribution to the total variance of the data and these can be discarded without major loss of information
- We can retain 90% variance of the data by storing the largest eigen values and eigen vectors which contribute 90% of the variance and projecting our data on these bases
- Proportion of Variance (PoV) explained

$$PoV(k) = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_k + \cdots + \lambda_d}$$

  when λ_i are sorted in descending order
- Typically, stop at PoV > 0.9
- Scree graph plots of PoV vs *k*, stop at "elbow"
- Now the *d*-dimensional data vector *x* with associated mean vector μ can be projected using the *k x d* dimensional *W* matrix containing the *k* selected eigen vectors to obtain a *k* < *d* dimensional data vector *z* using
  - $z = W^T(x - \mu)$


(a) Scree graph for Optdigits
Eigenvalues
Eigenvectors
0
100
200
0
10
20
30
40
50
60
70
(b) Proportion of variance explained
Prop of var
Eigenvectors
0
0.2
0.4
0.6
0.8
1


Optdigits after PCA
Second Eigenvector
First Eigenvector
30
20
10
0
−10
−20
−30
−40
−40
−30
−20
−10
0
10
20
30
40

Algorithm for PCA: Classical Method

- Each of the N samples is stored as a d-dimensional vector $x^i = \left[x_1^i \ \cdots \ x_d^i \right]^T$
- The data matrix is formed as $X = \left[x^1 \ \cdots \ x^N \right]$
- Compute the mean m=[m_1 … m_d]^T from *X* using $m_i = \frac{1}{N}\sum_{j=1}^{P} x_i^j$
- Centralize each sample in the data as $\bar{x}^i = x^i - m$ $\quad \bar{X} = \left[\bar{x}^1 \ \cdots \ \bar{x}^N \right]$
- Compute Covariance Matrix $S = \frac{\bar{X}\bar{X}^T}{N-1}$
- Find the Eigen Values $\lambda_1, \lambda_2 \ldots \lambda_d$ & *d*-dimensional Eigen Vectors $w_1, w_2 \ldots w_d$, of *S* using $S\lambda = w\lambda$ and sort the eigen values in decreasing magnitudes. Normalize the eigen vectors.
- Calculate the required dimension *k* based on proportion of variance based approach explained earlier for a given threshold
- Form $W = [\, w_1\, w_2 \ldots w_k \,]_{(d \times k)}$
- A vector *x* can be projected using $z = W^T(x - m)$

Algorithm for PCA: Snapshot Method

- If the input dimension (d) is large then the size of the covariance matrix is also large making its calculations computationally demanding
- It is known that for a d x N matrix the maximum number of non-zero eigenvectors is min(d-1,N-1)
- If N < d, then we can compute the eigen vectors w_i' of

$$S'_{(N\times N)} = \frac{\bar{X}^T X}{N-1}$$

instead of S. The eigen values for both S and S' are same and the eigen vectors of S can be obtained from those of S' using

$$w_{i_{(d\times 1)}} = \bar{X}_{(d\times N)} w'_{i\,(N\times 1)}$$

Algorithm for PCA: Snapshot Method

- Each of the N samples is stored as a d-dimensional vector $x^i = \begin{bmatrix} x_1^i & \cdots & x_d^i \end{bmatrix}^T$
- The data matrix is formed as $X = \begin{bmatrix} x^1 & \cdots & x^N \end{bmatrix}$
- Compute the mean $m=[m_1 \ldots m_d]^T$ from X using $m_i = \frac{1}{N}\sum_{j=1}^{P} x_i^j$
- Centralize each sample in the data as $\bar{x}^i = x^i - m$ $\bar{X} = \begin{bmatrix} \bar{x}^1 & \cdots & \bar{x}^N \end{bmatrix}$
- Compute Covariance Matrix $S' = \frac{\bar{X}^T \bar{X}}{N-1}$
- Find the Eigen Values (sorted in decreasing values) $\lambda'_1, \lambda'_2 \ldots \lambda'_d$ & *d*-dimensional Eigen Vectors $w'_1, w'_2 \ldots w'_d$, of S using $S\lambda'=w'\lambda'$.
- Calculate the required dimension *k* based on proportion of variance based approach explained earlier for a given threshold
- Form $W=[\, w_1\, w_2 \ldots w_k\,]_{(d \times k)}$ $w_i^* = \bar{X}w_i'$, $w_i = \frac{w_i^*}{|w_i^*|}$
- A vector *x* can be projected using $z = W^T(x- m)$

PCA in Scikit-learn

- http://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_iris.html