Machine Learning

Lecture 16: Principal Component Analysis (PCA)

Dr. Beilun Wang

Southeast University
School of Computer Science
and Engineering

This course

- Regression (supervised)
- Classification (supervised)
  - Feature selection
- Unsupervised models
  - Dimension Reduction (PCA)
  - Clustering (K-means, GMM/EM, Hierarchical)
- Learning theory
- Graphical models

	X_1	X_2	X_3
S_1			
S_2			
S_3			
S_4			
S_5			
S_6			

a data matrix of n observations on p variables $x_1, x_2, \ldots x_p$

Unsupervised learning = learning from raw (unlabeled, unannotated, etc) data, as opposed to supervised data where a classification/regression label of examples is given

- **Data**/*points/instances/examples/samples/records*: [rows]
- **Features**/*attributes/dimensions/independent variables/covariates/predictors/regressors*: [columns]

Today

- Dimensionality Reduction (unsupervised) with Principal Components Analysis (PCA)
    - Review of eigenvalue, eigenvector
    - How to project samples into a line capturing the variation of the whole dataset ⇨ Eigenvector / Eigenvalue of covariance matrix
    - PCA for dimension reduction
    - Extra: PCA examples

Review: Eigenvector/Eigenvalue

- The eigenvalues λ_i are found by solving the equation

$$\det(C - \lambda I) = 0$$

$$\mathrm{Cu} = \lambda \mathrm{u}$$
$$\mathrm{u} \neq 0$$

- Eigenvectors are columns of the matrix U such that

$$\mathrm{C} = \mathrm{UDU^T}$$

- Where $\mathrm{D}= \begin{pmatrix} \lambda_1 0 \ldots\ldots 0 \\ 0 \quad \lambda_2 \ldots\ldots 0 \\ 0 \\ 0 \ldots\ldots\ldots\ldots \lambda_p \end{pmatrix}$

Review: Eigenvalue, e.g.

- Let us take two variables with covariance *c>0*

- $C = \begin{pmatrix} 1 & c \\ c & 1 \end{pmatrix}$ $\quad C - \lambda I = \begin{pmatrix} 1-\lambda & c \\ c & 1-\lambda \end{pmatrix}$

$$\det(C - \lambda I) = (1-\lambda)^2 - C^2$$

$$Cu = \lambda u$$
$$u \neq 0$$

- Solving this we find $\lambda_1 = 1 + c$

$$\lambda_2 = 1 - c < \lambda_1$$

Review: Eigenvector, e.g.

- Any eigenvector U satisfies the condition

$$\mathrm{C}\mathrm{u} = \lambda \mathrm{u}$$

- $\mathrm{u} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ $\mathrm{Cu} = \begin{pmatrix} 1 & c \\ c & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} \lambda a_1 \\ \lambda a_2 \end{pmatrix}$

- After orthogonalization, we find

$$\mathrm{u}_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}, \mathrm{u}_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix}$$

Today

- Dimensionality Reduction (unsupervised) with Principal Components Analysis (PCA)
  - Review of eigenvalue, eigenvector
  - How to project samples into a line capturing the variation of the whole dataset ➡ Eigenvector / Eigenvalue of covariance matrix
  - PCA for dimension reduction
  - Extra: PCA examples

Background: Big & High-Dimensional Data

- High-Dimensions = Lot of Features

  Document classification

  Features per document =

  thousands of words/unigrams mi
  contextual information

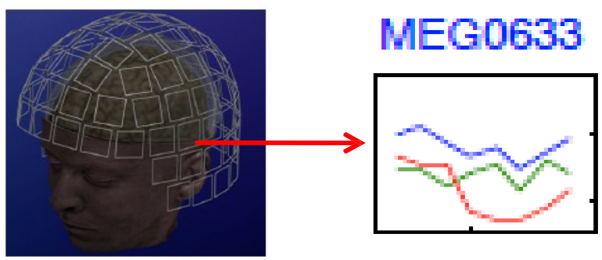
Surveys - Netflix

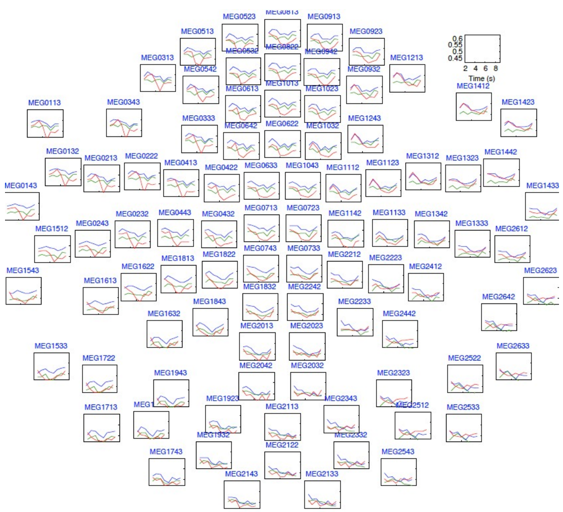
480189 users x 17770 movies

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6
Tom	5	?	?	1	3	?
George	?	?	3	1	2	5
Susan	4	3	1	?	5	1
Beth	4	3	?	2	4	2

Background: Big & High-Dimensional Data

- High-Dimensions = Lot of Features
- MEG Brain Imaging
  - 120 locations x 500 time points x 20 objects





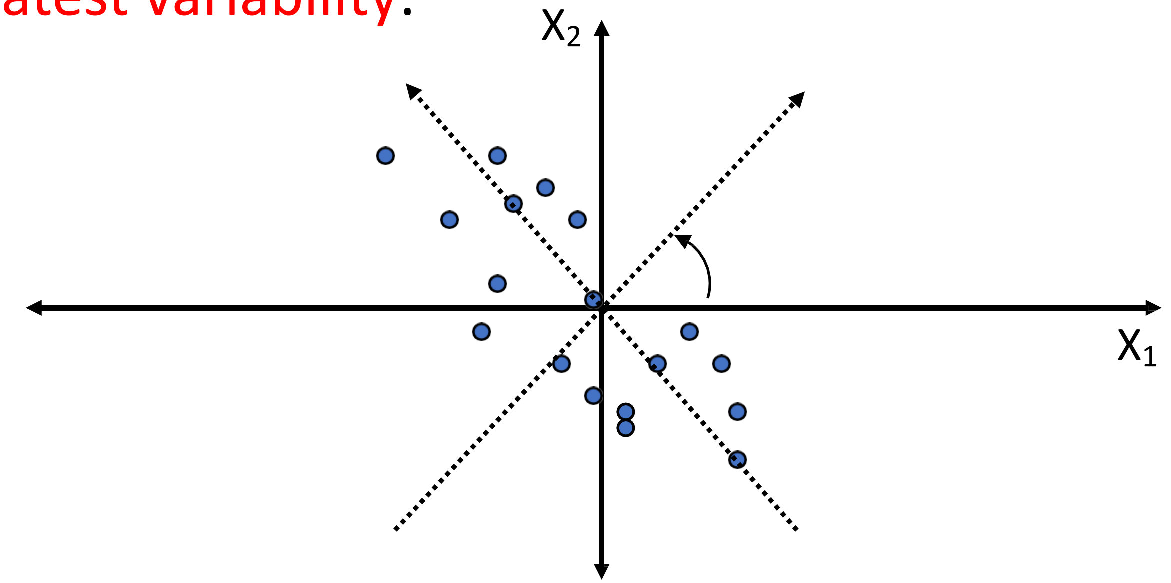
Or any high-dimensional image data

Goal

- We wish to explain/summarize the underlying variance-covariance structure of a large set of variables through a few linear combinations of these variables.

Trick: Rotate Coordinate Axes

- Suppose we have a sample population measured on p random variables $X_1, \ldots, X_p$.
- Our goal is to develop a new set of K (K<p) axes
- (linear combinations of the original p axes) in the directions of greatest variability:



This could be accomplished by rotating the axes (if data is centered).

Algebraic Interpretation

- Given n points in a p dimensional space,
- for large p, how to project on to a lower-dimensional ($K<p$) space while preserving broad trends in the data and allowing it to be visualized?

From now we assume Data matrix is centered: we subtract the mean along each dimension, and center the original axis system at the centroid of all data points, for simplicity.

Review: Projection

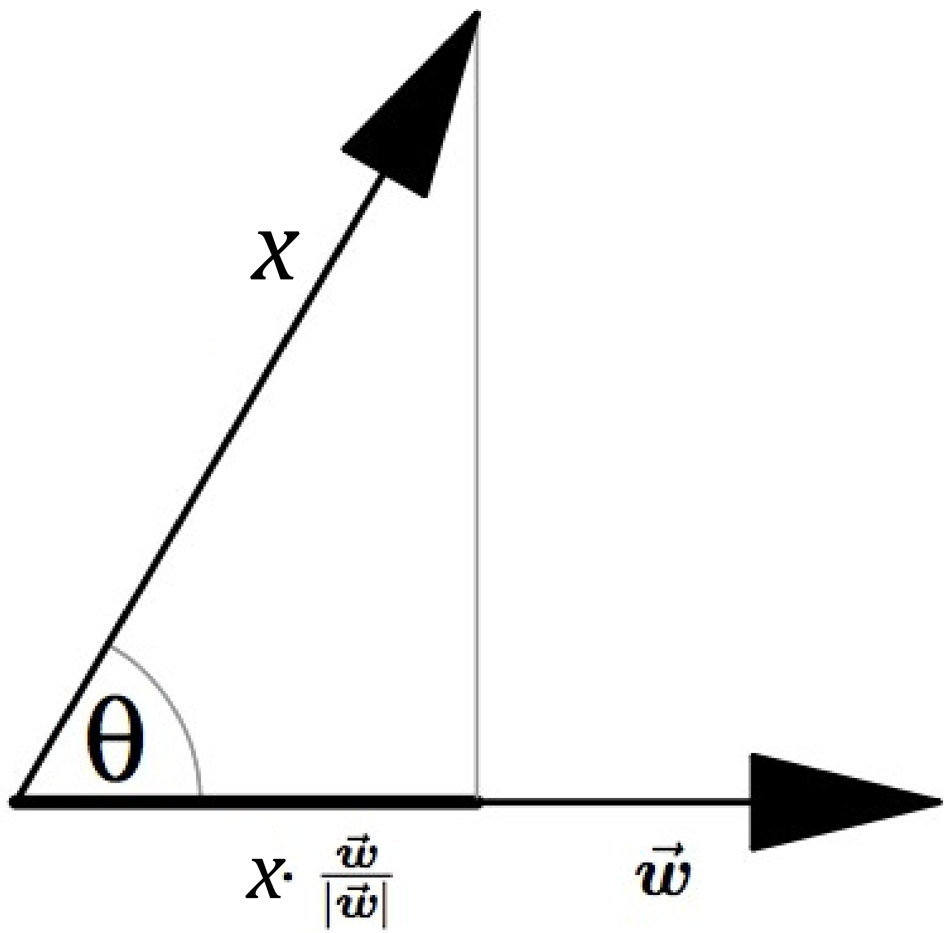
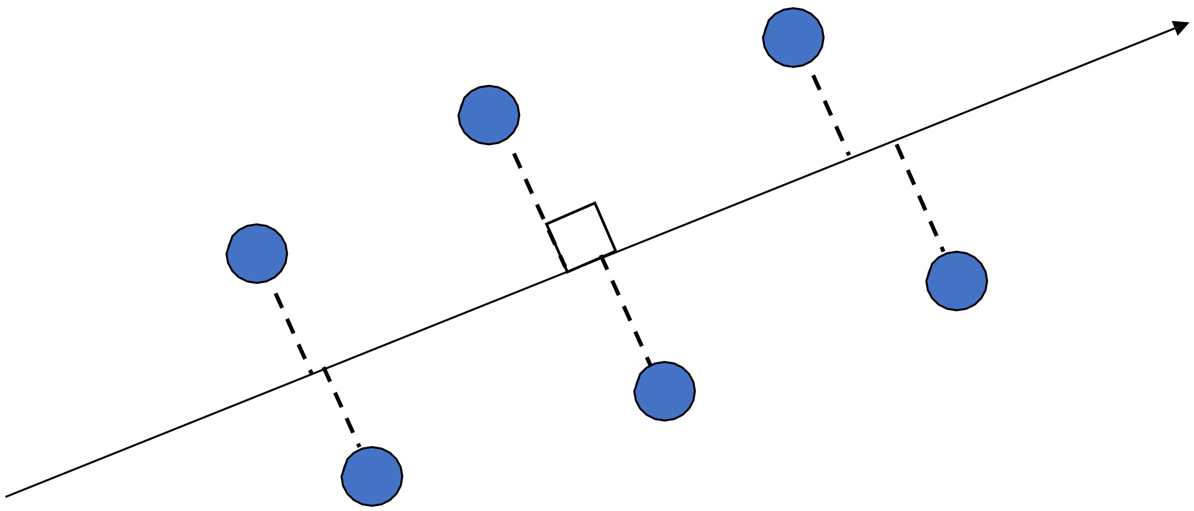


Figure 1: The dot product is fundamentally a projection.

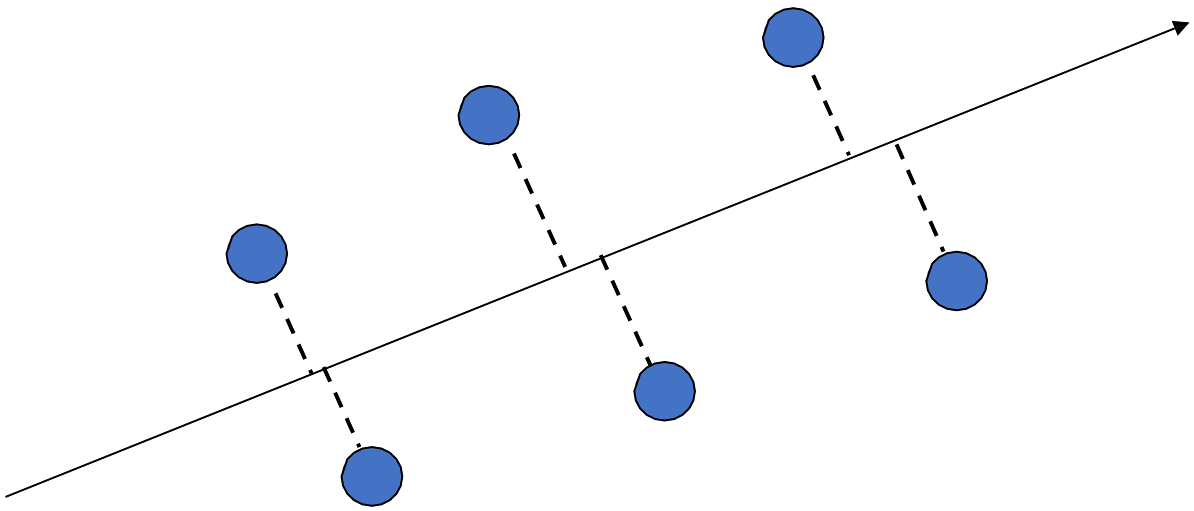
Algebraic Interpretation – 1D

- Given n points in a p dimensional space, how to project on to a 1 dimensional space?



Algebraic Interpretation – 1D

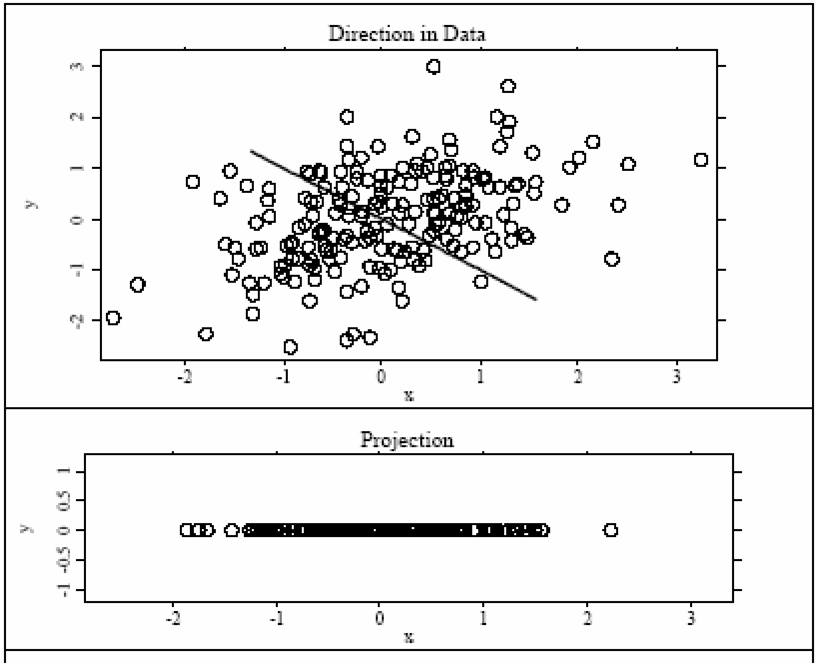
- Formally, to find a line that maximizing the sum of squares of data samples’ projections on that line



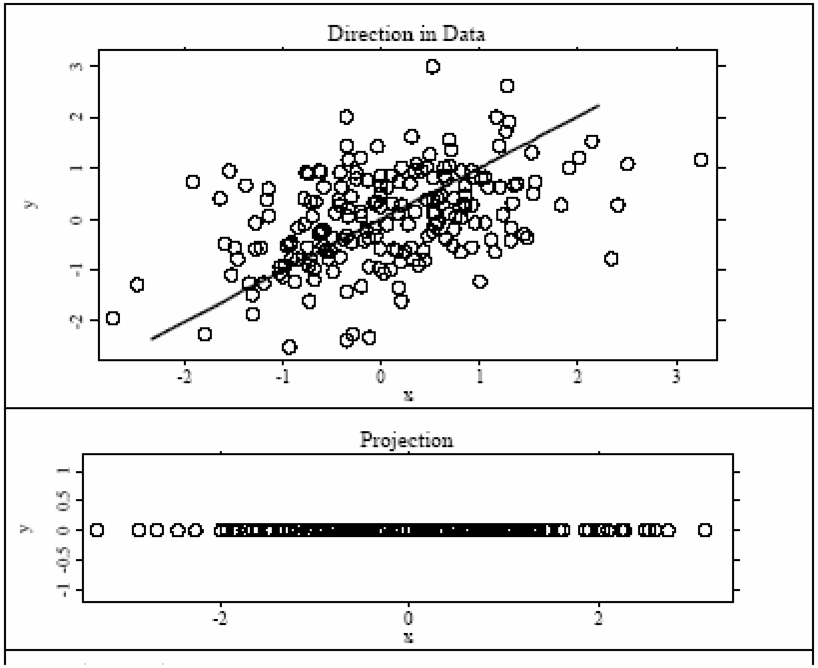
Algebraic Interpretation – 1D

- Example:

Good

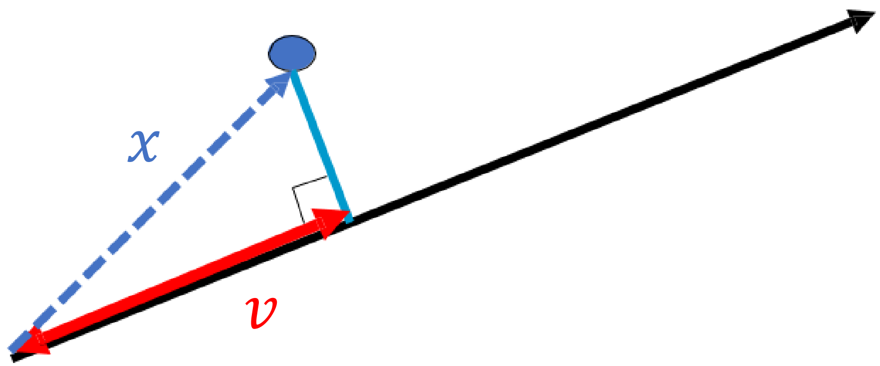


Better



Algebraic Interpretation – 1D

- Formally, to find a line (direction) that maximizing the sum of squares of data samples' projections on that line



$$u = x^T v$$

size of x's projection on vector v → $u = x^T v = v^T x$

subject to $v^T v = 1$

x: p*1 vector

v: p*1 vector

u: 1*1 scalar

Algebraic Interpretation – 1D

$$\max\left\{\sum_{i=1}^{n} u_i^2\right\}$$

$$\vec{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} x_1^T v \\ x_2^T v \\ \vdots \\ x_n^T v \end{bmatrix} = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{bmatrix} v = \underset{n \times p}{X}\ \underset{p \times 1}{v}$$

Algebraic Interpretation – 1D

- How is the sum of squares of projection lengths expressed in algebraic terms?

$$\max(v^T X^T X v)\,, \text{subject to } v^T v = 1$$

Algebraic Interpretation – 1D

- Rewriting:

$$v^T X^T X v = \lambda = \lambda v^T v = v^T(\lambda v)$$

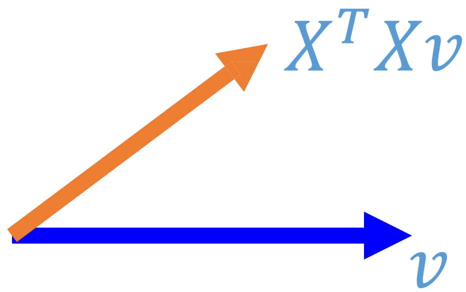
$$<=> \quad v^T(X^T X v - \lambda v) = 0$$

  - It shows that the maximum value of $v^T X^T X v$ is obtained for those vectors / directions satisfying $X^T X v = \lambda v$

  - So, find the largest λ and associated v such that the matrix $X^T X$ when applied to v, yields a new vector which is in the same direction as v, only scaled by a factor λ.

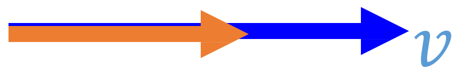
Algebraic Interpretation – 1D

- $X^T X v$ points in some other direction (different from v) in general



- If v is an eigenvector and λ is corresponding eigenvalue

$$X^T X v = \lambda v$$

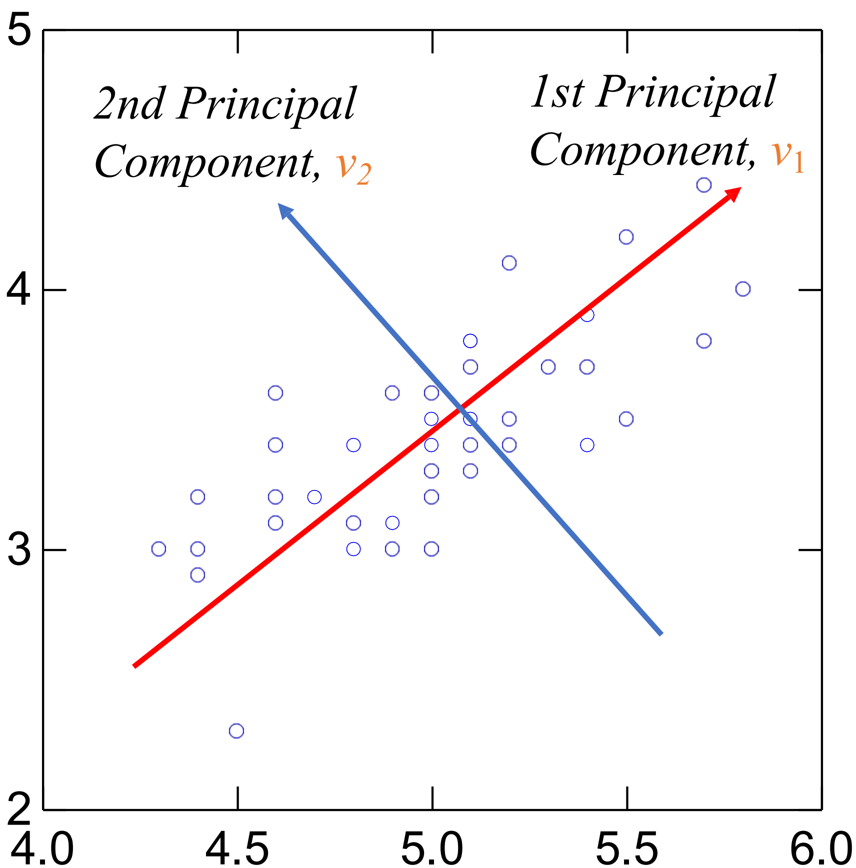


Algebraic Interpretation – beyond 1D

- For matrices of the form (symmetric) $X^T X$
  - All eigenvalues are non-negative
  - $\lambda_1, \ldots, \lambda_p$ are the eigenvalues, ordering from large to small
    - *i.e. Ordered by the PC's importance*

PCA Eigenvectors → Principal Components

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_p$$



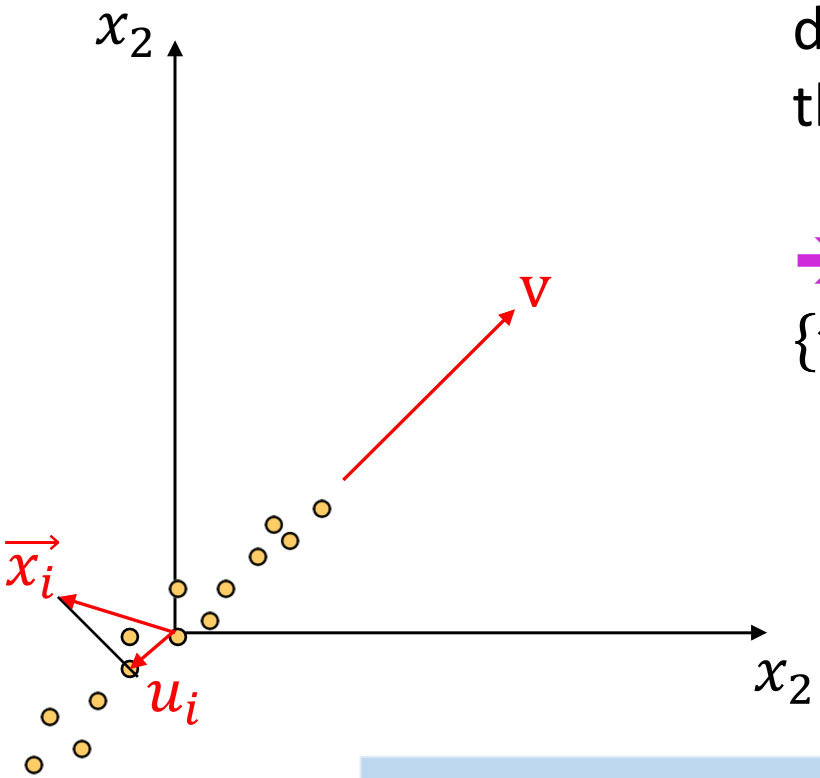
PCA (k=1): How the sum of squares of projection lengths relates to Variance?

size of x's projection on vector v

$\rightarrow u = x^T v = v^T x$

- In a new coordinate system with v as axis, u is the position of sample x on this axis

PCA (k=1): How the sum of squares of projection lengths relates to Variance?



Consider the variation along direction $\boldsymbol{v}$ considering all of the points $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$

➔The variance of all positions $\{u_1, u_2, \ldots, u_n\}$

convert $\boldsymbol{x}_i$ onto $\boldsymbol{v}$ coordinate

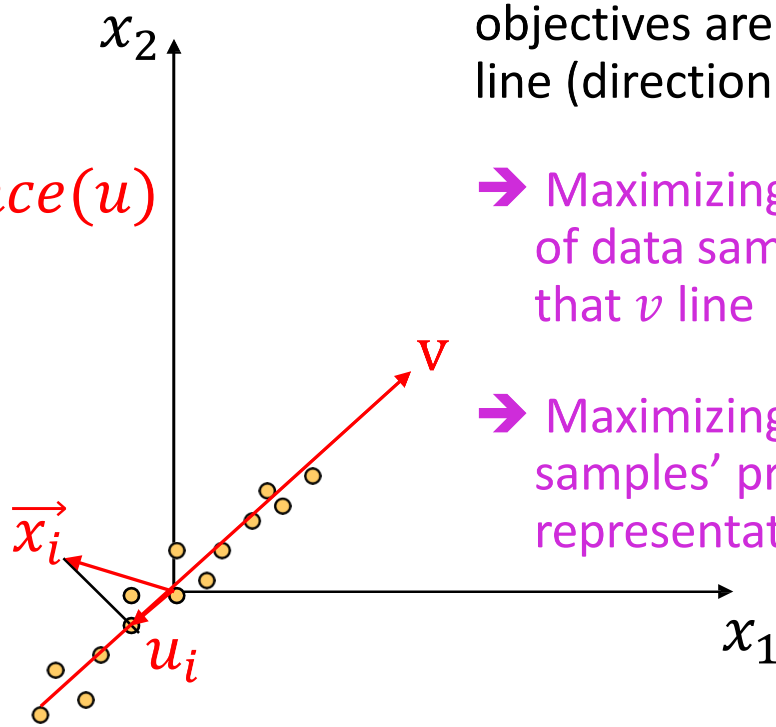
➔ $u_i = \boldsymbol{x}_i^T \boldsymbol{v}$

How the sum of squares of projection lengths relates to Variance?

$$Var(u) = \sum_{u_i} (u_i - \mu)^2 \mathrm{P}(u = u_i) = \sum_{u_i} (u_i)^2$$

Assuming centered data matrix

$$\underset{v}{argmax} \sum u_i^2$$
$$= \underset{v}{argmax}\, Variance(u)$$



This means the following two objectives are the same, for finding a line (direction v) by

- ➔ Maximizing the sum of squares of data samples' projections on that v line
- ➔ Maximizing the variance of data samples' projected representations on the v axis

Today

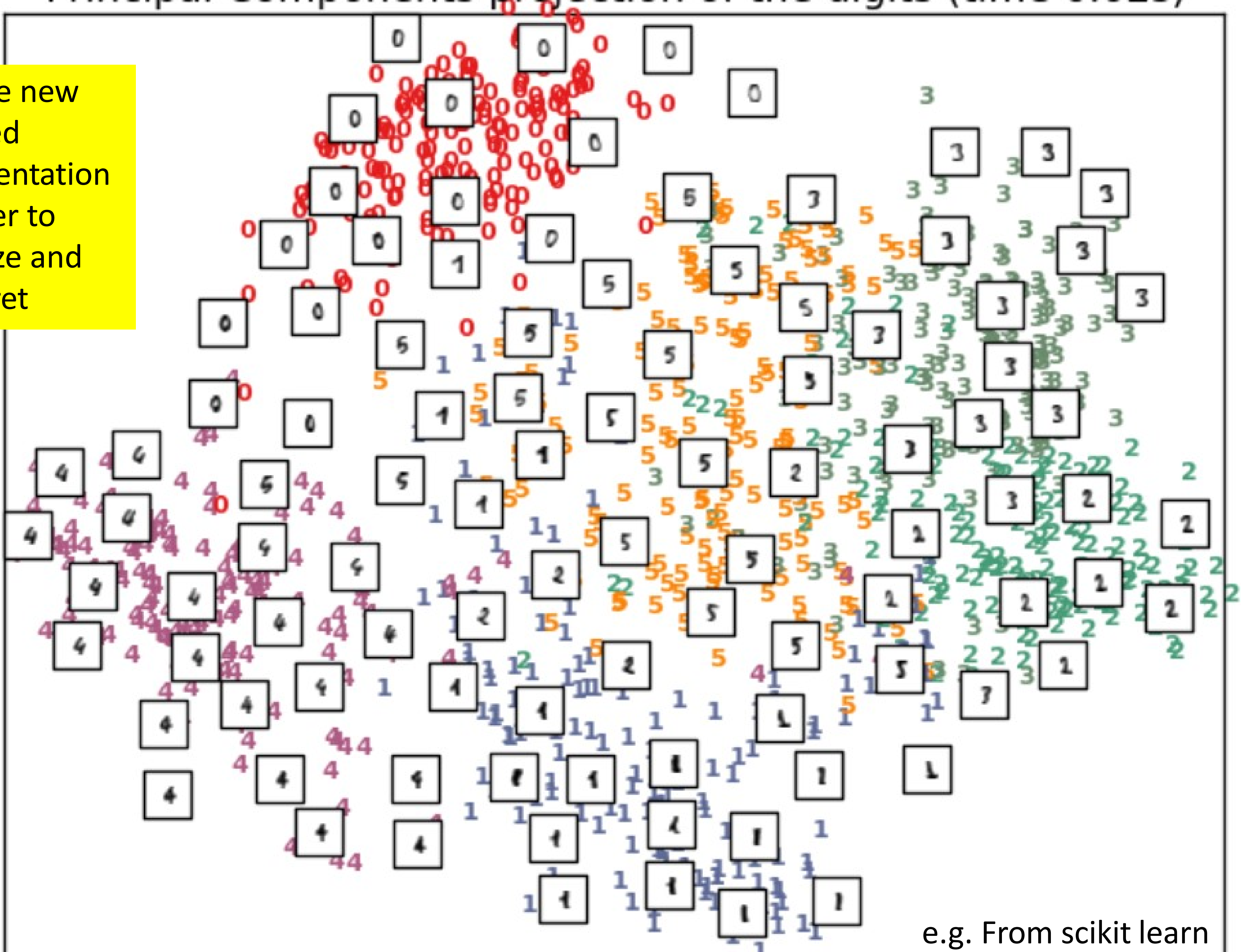
- Dimensionality Reduction (unsupervised) with Principal Components Analysis (PCA)
  - Review of eigenvalue, eigenvector
  - How to project samples into a line capturing the variation of the whole dataset ⇒ Eigenvector / Eigenvalue of covariance matrix
  - PCA for dimension reduction
  - Extra: PCA examples

Applications

- Uses:
  - Data Visualization
  - Data Reduction
  - Data Classification
  - Trend Analysis
  - Factor Analysis
  - Noise Reduction

- Examples:
  - How many unique “sub-sets” are in the sample?
  - How are they similar / different?
  - What are the underlying factors that influence the samples?
  - How to best present what is “interesting”?
  - Which “sub-set” does this new sample rightfully belong?
  - ....

Principal Components projection of the digits (time 0.02s)



e.g. the new reduced representation is easier to visualize and interpret

e.g. From scikit learn

Interpretation of PCA

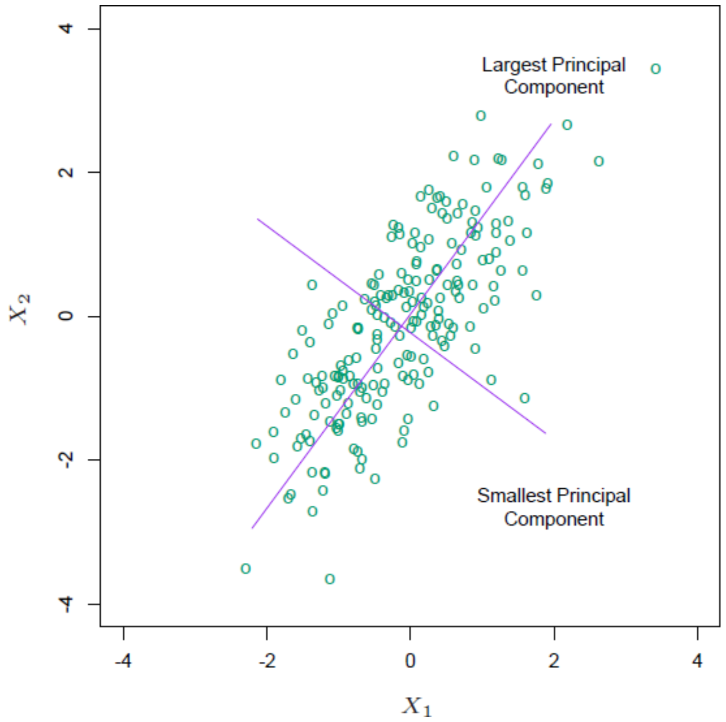
- From p original coordinates: $x_1, x_2, \ldots, x_p$:
- Produce k new coordinates : $v_1, v_2, \ldots, v_k$:

When p=2

$$v_1 = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1p}x_p$$

$$v_2 = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2p}x_p$$

…



Interpretation of PCA

- v_k's are Principal Components
  - v_k are uncorrelated (orthogonal) from each other
  - v_1 explains as much as possible of original variance in data set
  - v_2 explains as much as possible of remaining variance etc.
  - v_k : k^{th} PC retains the k^{th} greatest fraction of the variation in the samples

Interpretation of PCA

- The new variables (PCs) have a variance equal to their corresponding eigenvalue, since

$$Var(u^k) = v_k^T X^T X v_k = v_k^T \lambda_k v_k = \lambda_k v_k^T v_k = \lambda_k$$

for all $k = 1, \dots, p$

- Small λ_k ⇔ small variance ⇔ data change little in the direction of component v_k

PCA is useful for finding new, more informative, uncorrelated features; it reduces dimensionality by rejecting low variance features

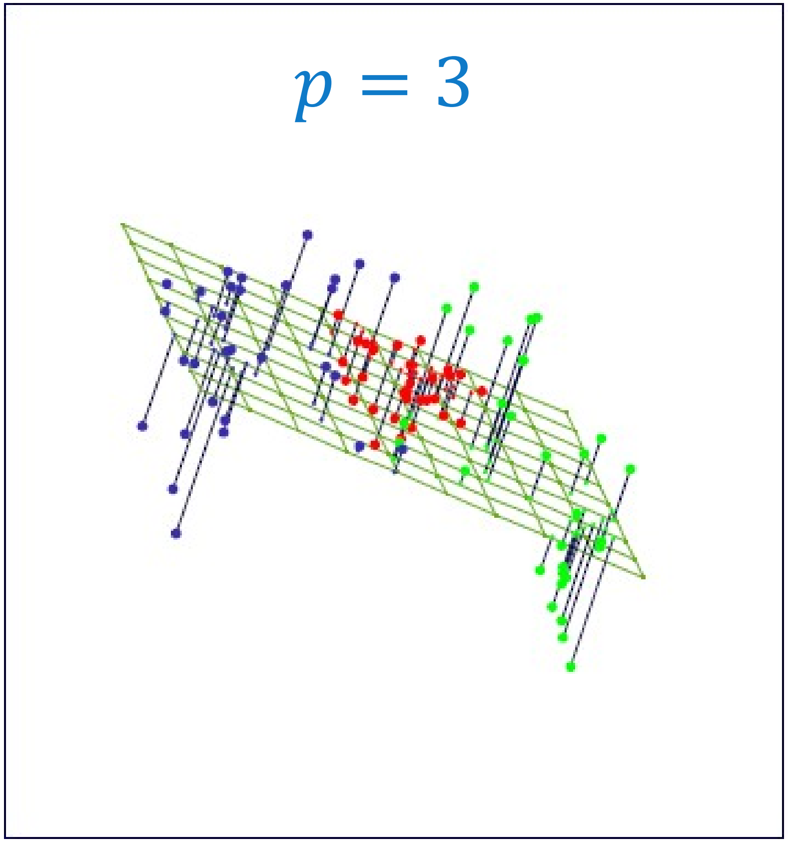
PCA Summary until now

- Rotates multivariate dataset into a new configuration which is easier to interpret

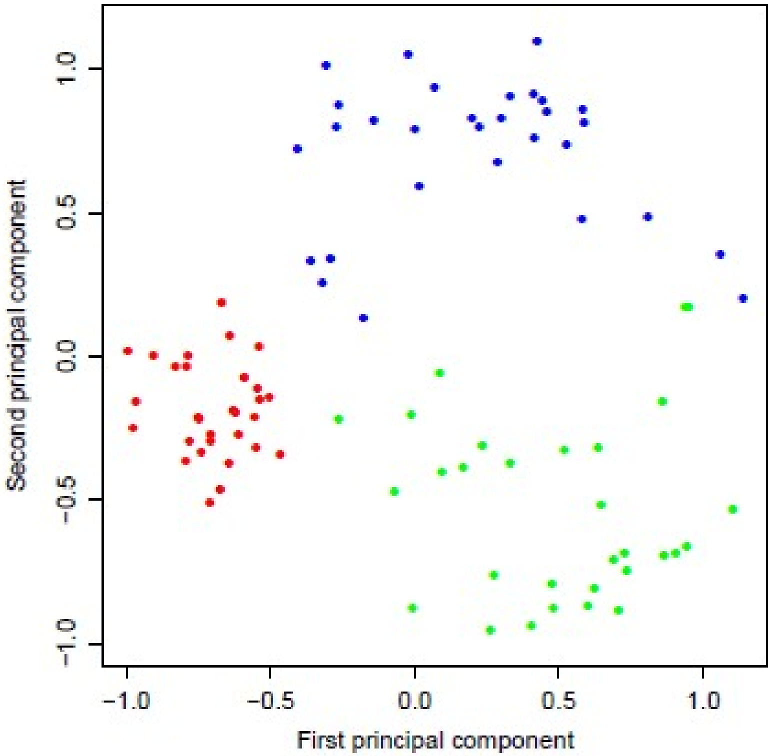
- PCA is useful for finding new, more informative, uncorrelated features; it reduces dimensionality by rejecting low variance features

> - PCA compresses (i.e. perform projection) the data points by only using the top few eigenvectors.
> - This corresponds to choosing a “linear subspace” represent points on a line, plane, or “hyper-plane”

PCA for dimension reduction
e.g. p=3 ➔ (pick top k=2 PCs)



$k = 2$



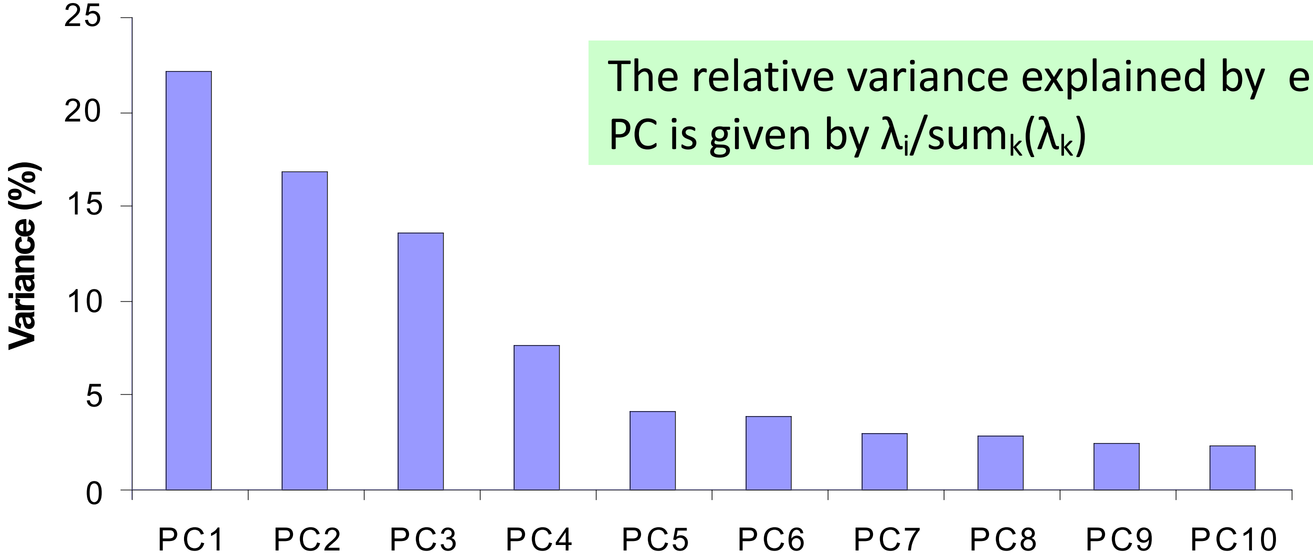
corresponds to choosing a "2D linear plane"

How many components to keep?

- I. Variance: Enough PCs to have a cumulative variance explained by the PCs that is >50-70%

- II. Scree plot: represents the ability of PCs to explain the variation in data, e.g. keep PCs with eigenvalues >1

e.g. check percentage of kept variance

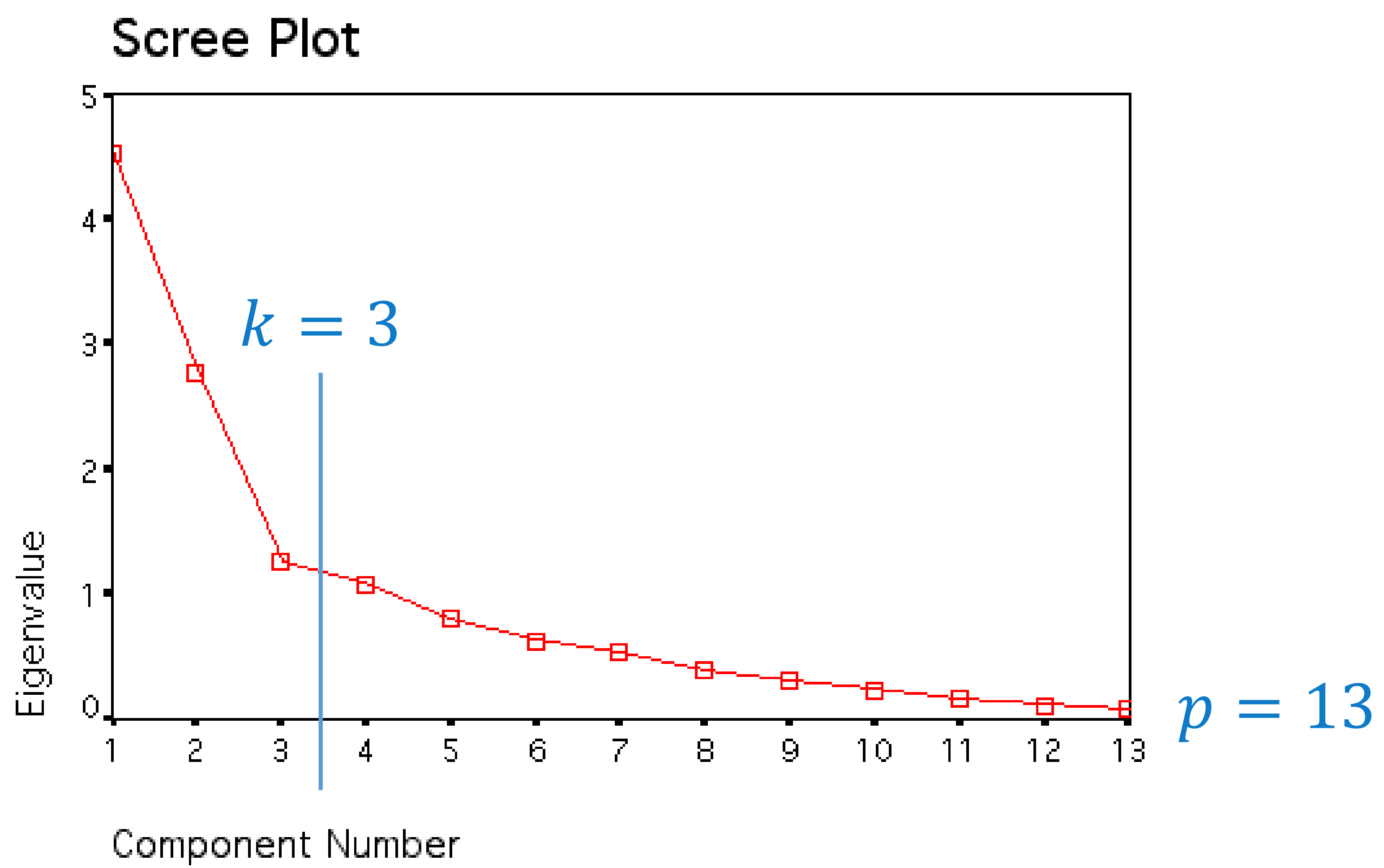


The relative variance explained by each PC is given by $\lambda_i/\mathrm{sum}_k(\lambda_k)$

You do lose some information, but if the eigenvalues are small, you don't lose much

- p dimensions in original data
- Calculate p eigenvectors and eigenvalues
- choose only the first k eigenvectors, by keep enough variance
- final projected data set has only k dimensions

e.g. check eigenvalue



Limitations of PCA

- PCA is not effective for some datasets.
- For example, if the data is a set of strings
- $(1,0,0,0,\ldots), (0,1,0,0\ldots), \ldots, (0,0,0,\ldots,1)$ then the eigenvalues do not fall off as PCA requires.

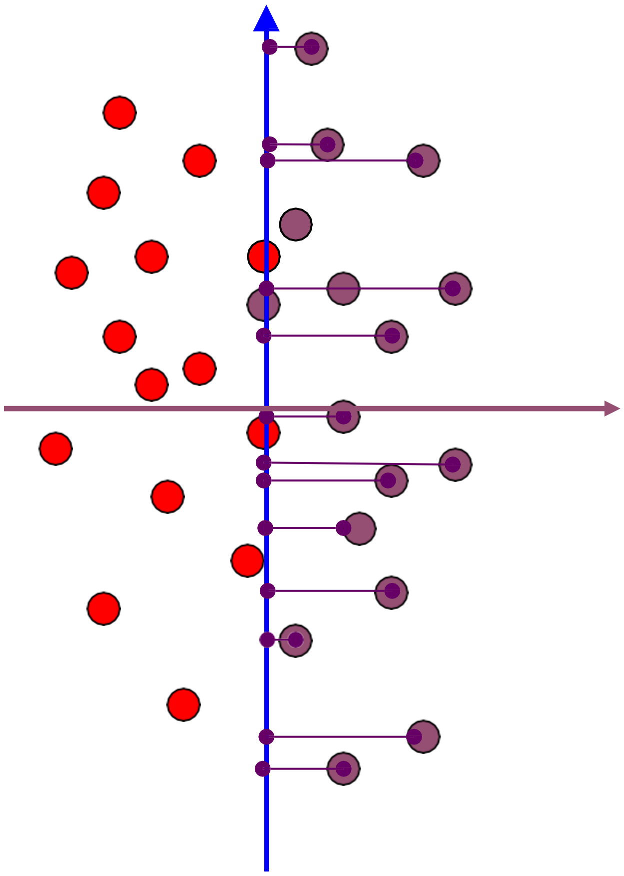
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad eigenvalue = [1,1,1]$$

Limitations of PCA

- The direction of maximum variance is not always good for classification (Example 1)

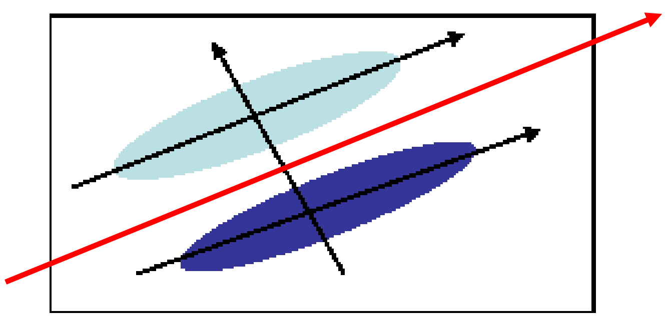
For this case:

- Ideal for capturing global variance
- Not ideal for discrimination

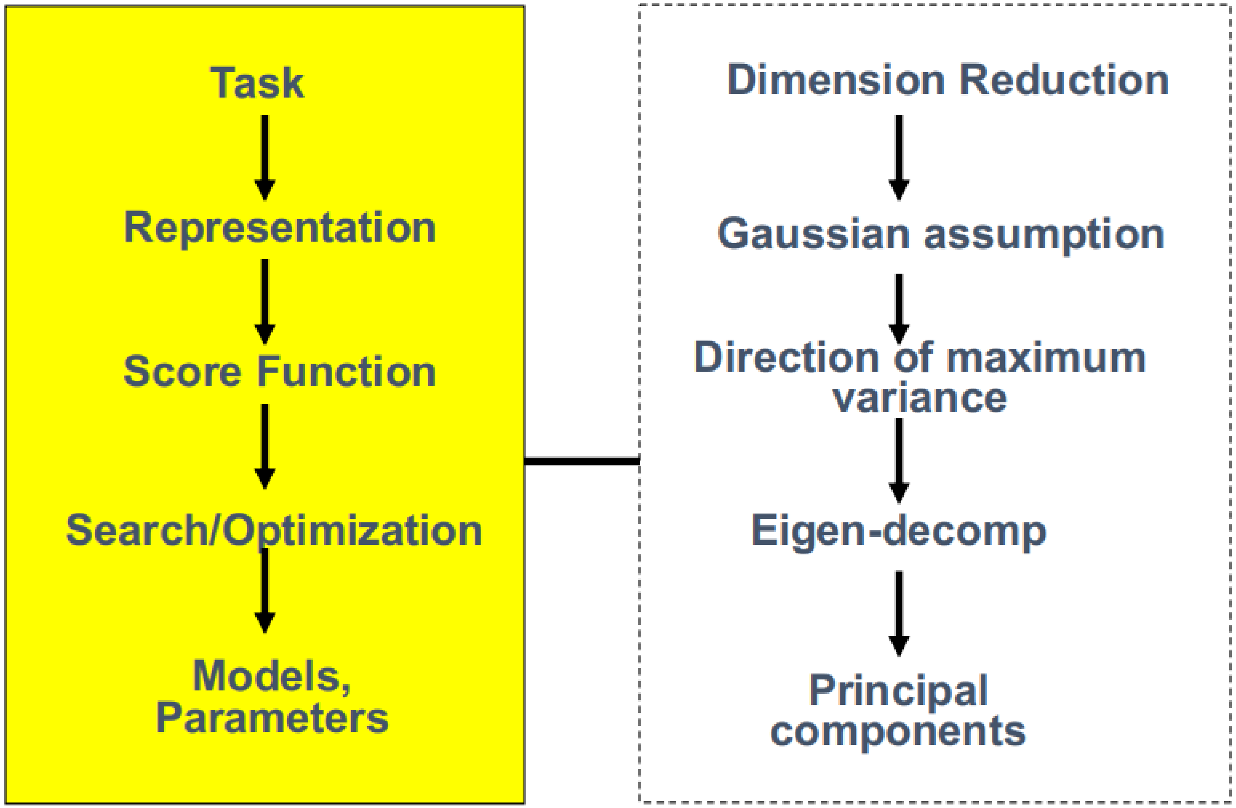


Limitations of PCA

- PCA may not find the best directions for discriminating between two classes. (Example 2)
- Example:
  - suppose the two classes have 2D Gaussian densities as ellipsoids.
  - 1st eigenvector is best for representing the probabilities / overall data trend
  - 2nd eigenvector is best for discrimination.



Principal Component Analysis



References

- Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No.1 New York: Springer, 2009.
- Dr. S. Narasimhan’s PCA lectures
- Prof. Derek Hoiem’s eigenface lecture

Today

- Dimensionality Reduction (unsupervised) with Principal Components Analysis (PCA)
  - Review of eigenvalue, eigenvector
  - How to project samples into a line capturing the variation of the whole dataset ➡ Eigenvector / Eigenvalue of covariance matrix
  - PCA for dimension reduction
  - Extra: PCA examples

Example: A 2D Numerical example

- Step 1: subtract the mean from each of the data dimensions.
  - It makes variance and covariance calculation easier by simplifying their equations.
  - The variance and covariance values are not affected by the mean value.

DATA: (p=2)

x1	x2
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

ZERO MEAN DATA:

x1	x2
.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

Example: A 2D Numerical example

- Step 2: calculate the covariance matrix

$$\text{cov} = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

- since the non-diagonal elements in this covariance matrix are positive, we should expect that the x1 and x2 variable increase together.

Example: A 2D Numerical example

- Step 3: calculate the eigenvectors and eigenvalues of the covariance matrix

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

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

Example: A 2D Numerical example

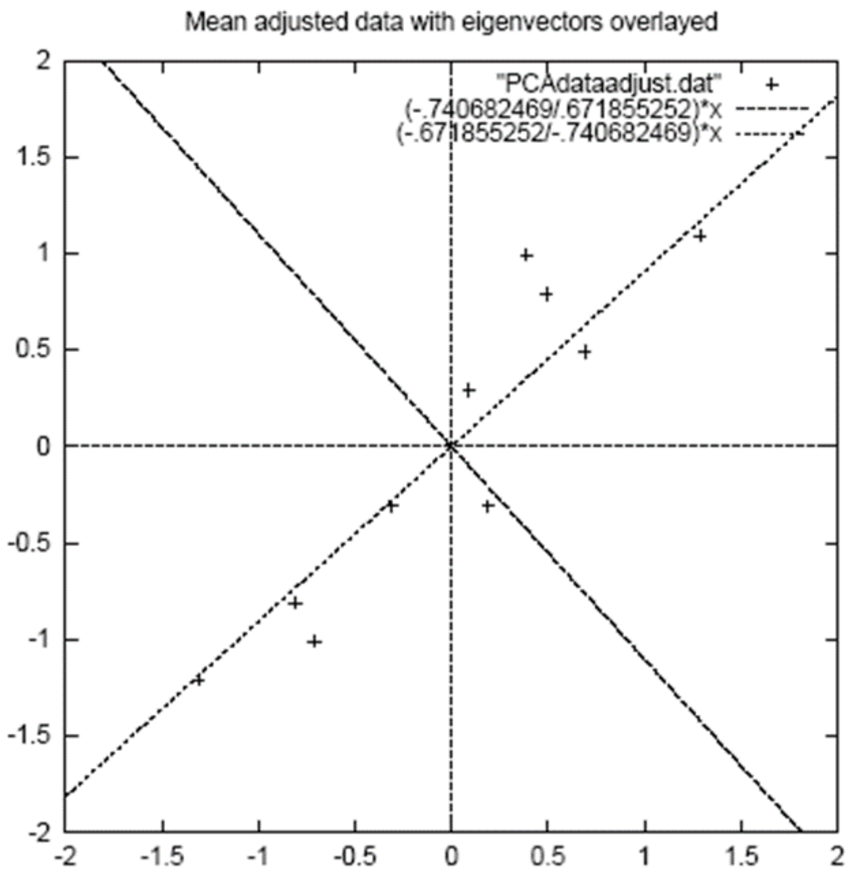


Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

- 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

Example: A 2D Numerical example

- Step 4: reduce dimensionality and form *feature vector*
- Feature Vector = $(eig_1\ eig_2\ eig_3 \ldots eig_n)$
- We can either form a feature vector with both of the eigenvectors:

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

or, we can choose to leave out the smaller, less significant component and only have a single column:

$$\begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

Example: A 2D Numerical example

- Step 5: derive the new data
  - FinalData = RowFeatureVector x RowZeroMeanData

  - 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

  - RowZeroMeanData is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

Example: A 2D Numerical example

<u>FinalData transpose: dimensions along columns</u>

w1	w2
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

Example: A 2D Numerical example

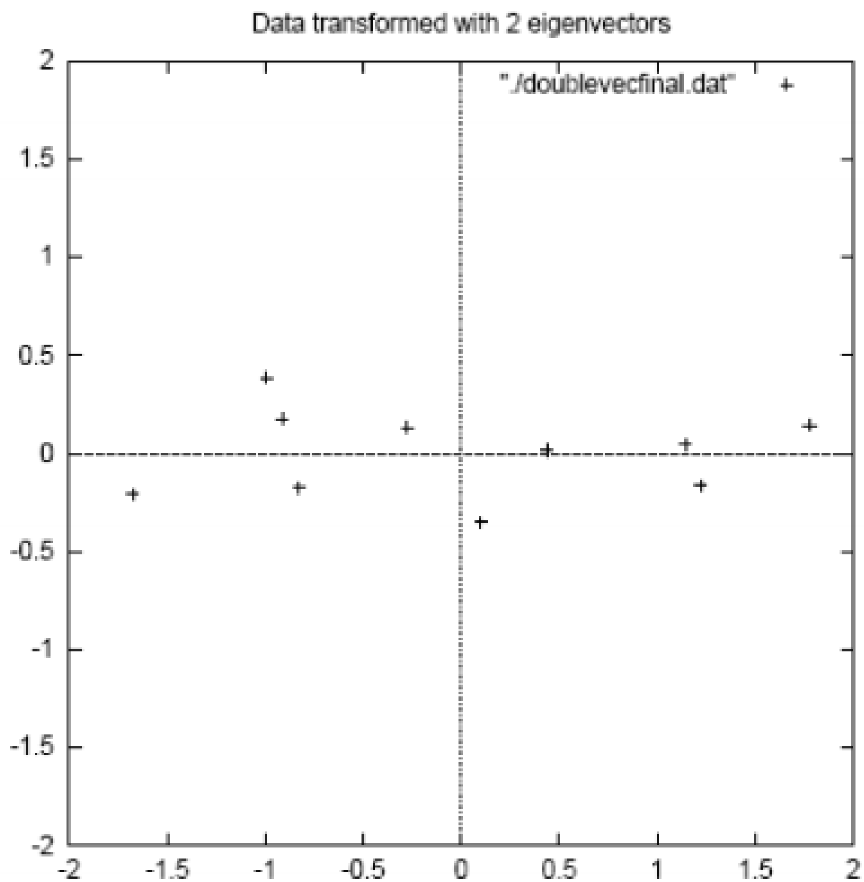


Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

Example: A 2D Numerical example

- Reconstruction of original Data
  - If we reduced the dimensionality, obviously, when reconstructing the data we would lose those dimensions we chose to discard.
  - In our example let us assume that we considered only the w1 dimension…

Example: A 2D Numerical example

w1

-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

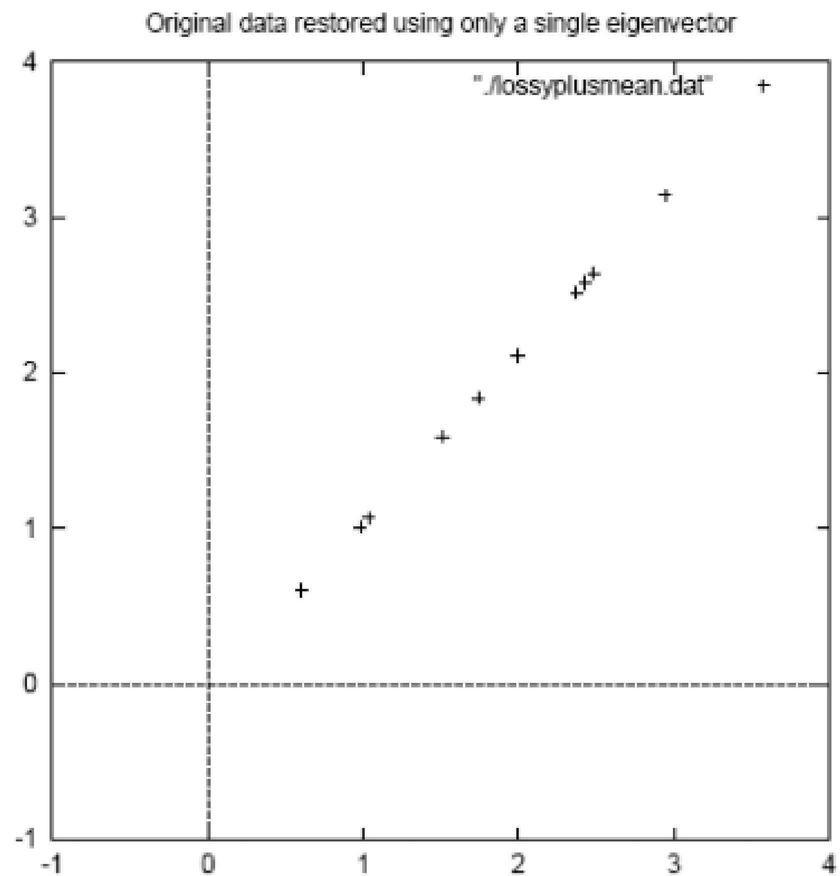


Figure 3.5: The reconstruction from the data that was derived using only a single eigenvector