# Feature extraction

Christian Salvatore
Scuola Universitaria Superiore IUSS Pavia

christian.salvatore@iusspavia.it

# FEATURE EXTRACTION

# Feature Extraction | Aim

Map from the initial space to a space smaller than $R^k$, with k <d

Operate in spaces of lower dimensionality:

- makes it easier to train machine-learning algorithms (requires less data for training)
- discarding redundant (correlated information) and noisy data also improves performance and makes it more robust

The goal is to discard information that is not relevant or less relevant to the problem of interest

Reducing dimensionality does not mean discarding some dimensions and saving others, but combining sizes appropriately.

# Feature Extraction | Main Techniques

Principal Component Analysis (PCA)
Unsupervised transformation that performs a linear mapping of dimensions with the aim of preserving the pattern information as much as possible (also known as Karhunen-Loeve transform)

# Feature Extraction | Principal Components Analysis

Given a training set $\mathbf{x}_i \in \Re^d, \quad i = 1...n$ , where

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1...n} \mathbf{x}_i$$

is the mean vector $\in \Re^d$

$$\boldsymbol{\Sigma} = \frac{1}{n-1} \sum_{i=1...n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^t$$

is the covariance matrix $\in \Re^{d \times d}$

then, for a given $k \ (k < d, k < n, k > 0)$, the k-dimensional space $(S_{\mathbf{x}}, \Phi_k)$

is univocally determined by the mean vector and the projection matrix $\Phi_k \in \Re^{d \times k}$

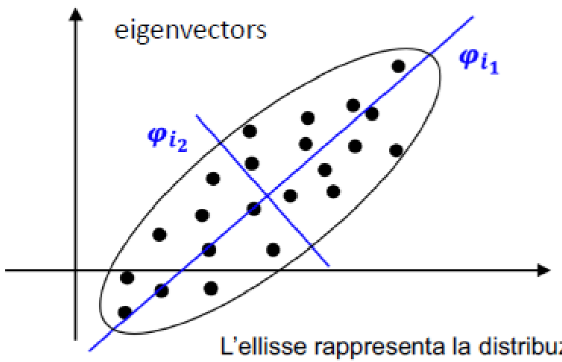
whose columns are made by the eigenvectors of Sigma corresponding to the highest k eigenvalues

$\Phi_k = [\boldsymbol{\varphi}_{i_1}, \boldsymbol{\varphi}_{i_2} ... \boldsymbol{\varphi}_{i_k}]$ con $\lambda_{i_1} \geq \lambda_{i_2} \geq \cdots \lambda_{i_k} \geq \cdots \lambda_{i_d}$

$\boldsymbol{\varphi}_{ir}$ eigenvector of Sigma corresponding to eigenvalue $\lambda_{ir}$ $r = 1...d$

**The first k eigenvectors are called PRINCIPAL COMPONENTS (PC)**

$\boldsymbol{\varphi}_{i1}$ **indicates the direction of highest variance in the (training) set**




L'ellisse rappresenta la distribuzione dei pattern nel training set

# Feature Extraction | Principal Components Analysis

Variance of one variable $X$:

$$\mathrm{Var}(X) = \frac{1}{n}\sum_j (\bar{x} - x_j)^2 = \sigma_X^2$$

Covariance of two variables $X$ and $Y$:

$$\mathrm{Cov}(X, Y) = \frac{1}{n}\sum_j (\bar{x} - x_j)(\bar{y} - y_j) = \sigma_{XY}$$

Covariance matrix of $n$ variables $X_1 \ldots X_n$:

$$\mathrm{C} = \begin{pmatrix} \sigma_{11}^2 & \sigma_{12} & \ldots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22}^2 & \ldots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \ldots & \sigma_{nn}^2 \end{pmatrix}$$

PCA diagonalizes the covariance matrix C:

$$\mathrm{C} = \mathrm{UDU^T}$$
$$= \mathrm{U} \begin{pmatrix} \lambda_1^2 & 0 & \ldots & 0 \\ 0 & \lambda_2^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n^2 \end{pmatrix} \mathrm{U^T}$$

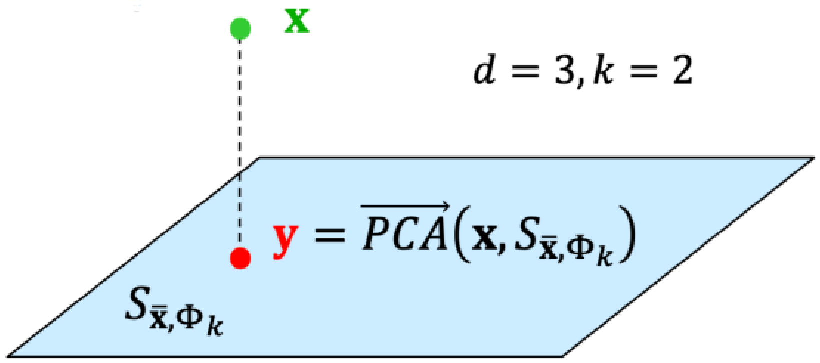
U: rotation matrix

D: diagonal matrix

$\lambda_j^2$: eigenvalues (= variance explained by each component)

Credits: Dimension reduction 1, Claus O. Wilke

# Feature Extraction | Principal Components Analysis




$\Re^d \rightarrow \Re^k$

**Proiezione** Una volta determinato lo spazio PCA, la proiezione di un pattern x su tale spazio è semplicemente la proiezione geometrica del vettore **x** sull'iperpiano che definisce lo spazio. In realtà la vera proiezione geometrica è un vettore che ha la stessa dimensionalità del vettore originale mentre in questo contesto indichiamo con proiezione il vettore (ridotto) nello spazio PCA. Matematicamente questa operazione è eseguita come prodotto della matrice di proiezione trasposta per il pattern x al quale è preventivamente sottratta la media.

$$\overrightarrow{PCA}(\mathbf{x}, S_{\bar{\mathbf{x}},\Phi_k}) = \Phi_k{}^t(\mathbf{x} - \bar{\mathbf{x}})$$

$\Re^k \rightarrow \Re^d$

**Retroproiezione** Dato un vettore **y** nello spazio PCA, la sua retro-proiezione verso lo spazio originale si ottiene moltiplicando il vettore per la matrice di proiezione e sommando il vettore medio. Questa trasformazione non sposta spazialmente il vettore, che giace ancora sullo spazio PCA, ma opera un cambiamento di coordinate che ne permette la codifica in termini delle $d$ componenti dello spazio originale.

$$\overleftarrow{PCA}(\mathbf{y}, S_{\bar{\mathbf{x}},\Phi_k}) = \Phi_k\mathbf{y} + \bar{\mathbf{x}}$$

# Feature Extraction | Principal Components Analysis

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}$$

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

$$s = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{(n-1)}}$$

**Eigenvectors**

**Eigenvalues**

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

$$var(X) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(X_i - \bar{X})}{(n-1)}$$

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

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

**Step 5: Choosing components and forming a feature vector**

**Step 5: Deriving the new data set**

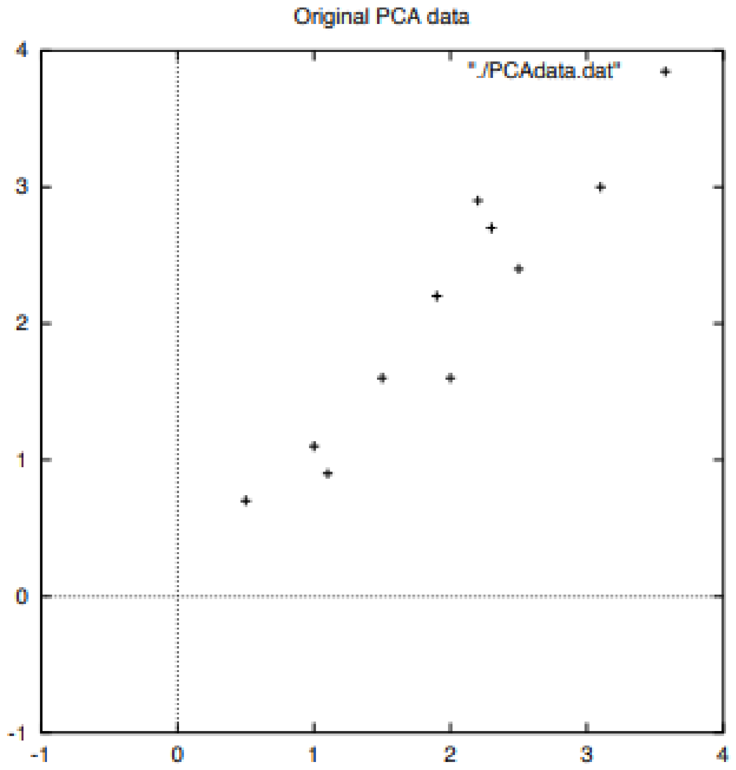
A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

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 |




A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

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 |

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 |

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

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 |

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

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

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 |

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

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

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

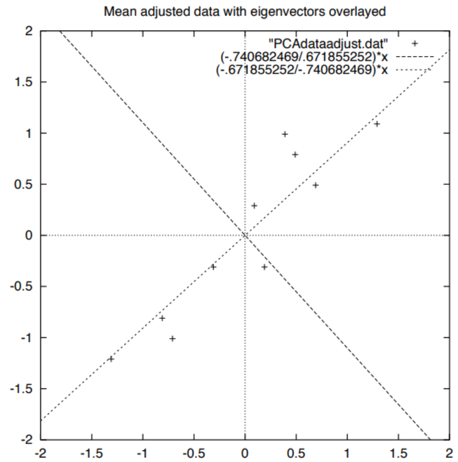



Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

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 |

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

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

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

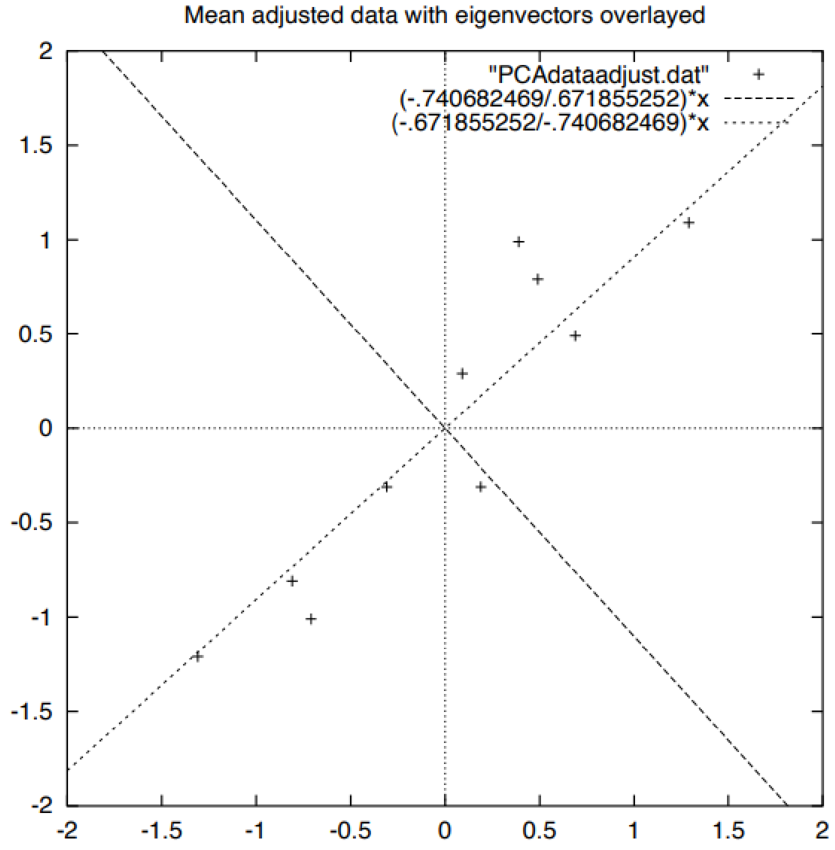



Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

**Step 5: Choosing components and forming a feature vector**

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 |

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

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

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

$$FeatureVector = (eig_1\ eig_2\ eig_3\ ....\ eig_n)$$

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

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

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

**Step 5: Choosing components and forming a feature vector**

**Step 5: Deriving the new data set**

$$FinalData = RowFeatureVector \times RowDataAdjust,$$

where $RowFeatureVector$ 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, and $RowDataAdjust$ is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

**Step 5: Choosing components and forming a feature vector**

**Step 5: Deriving the new data set**

$$FinalData = RowFeatureVector \times RowDataAdjust,$$

where $RowFeatureVector$ 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, and $RowDataAdjust$ is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

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 |

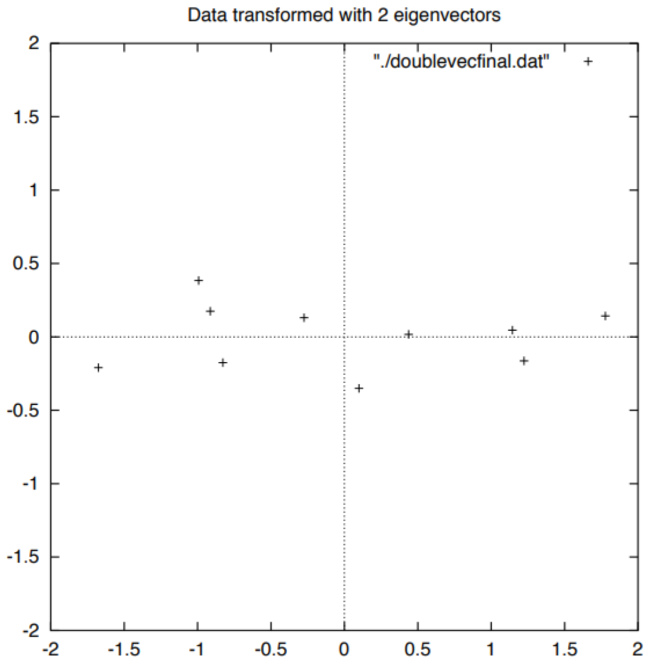



Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

**Step 5: Choosing components and forming a feature vector**

**Step 5: Deriving the new data set**

$$FinalData = RowFeatureVector \times RowDataAdjust,$$

where $RowFeatureVector$ 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, and $RowDataAdjust$ is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

Transformed Data (Single eigenvector)

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

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis

**Step 1: Get some data**

**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

**Step 5: Choosing components and forming a feature vector**

**Step 5: Deriving the new data set**

**Getting the old data back**

$$FinalData = RowFeatureVector \times RowDataAdjust$$

$$RowDataAdjust = RowFeatureVector^{-1} \times FinalData$$

$$RowDataAdjust = RowFeatureVector^{T} \times FinalData$$

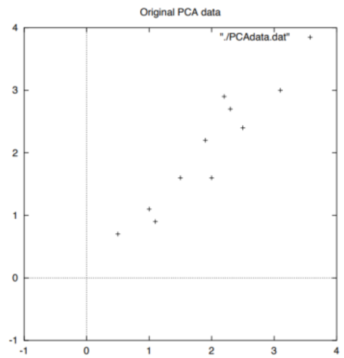
only true if the elements of the matrix are all unit eigenvectors

$$RowDataAdjust = RowFeatureVector^{T} \times FinalData$$

$$RowOriginalData = (RowFeatureVector^{T} \times FinalData) + OriginalMean$$

A tutorial on Principal Components Analysis by Lindsay I. Smith

# Feature Extraction | Principal Components Analysis




**Step 1: Get some data**

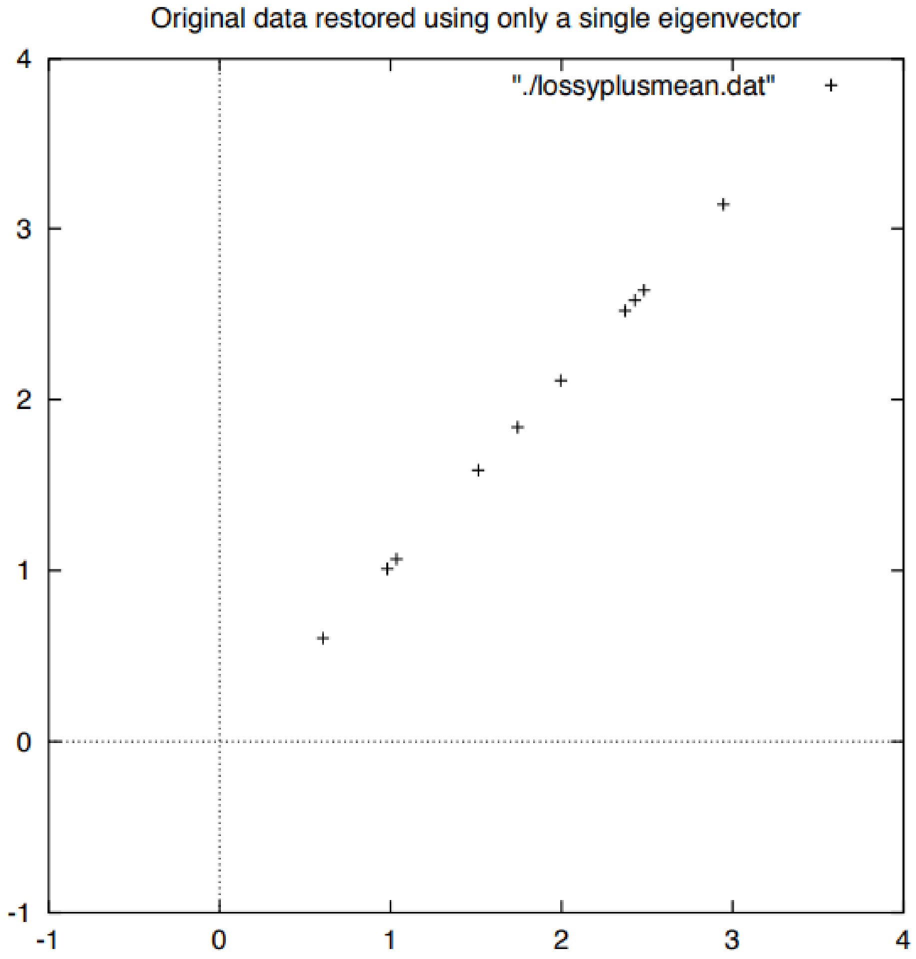
**Step 2: Subtract the mean**

**Step 3: Calculate the covariance matrix**

**Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**

**Step 5: Choosing components and forming a feature vector**

**Step 5: Deriving the new data set**

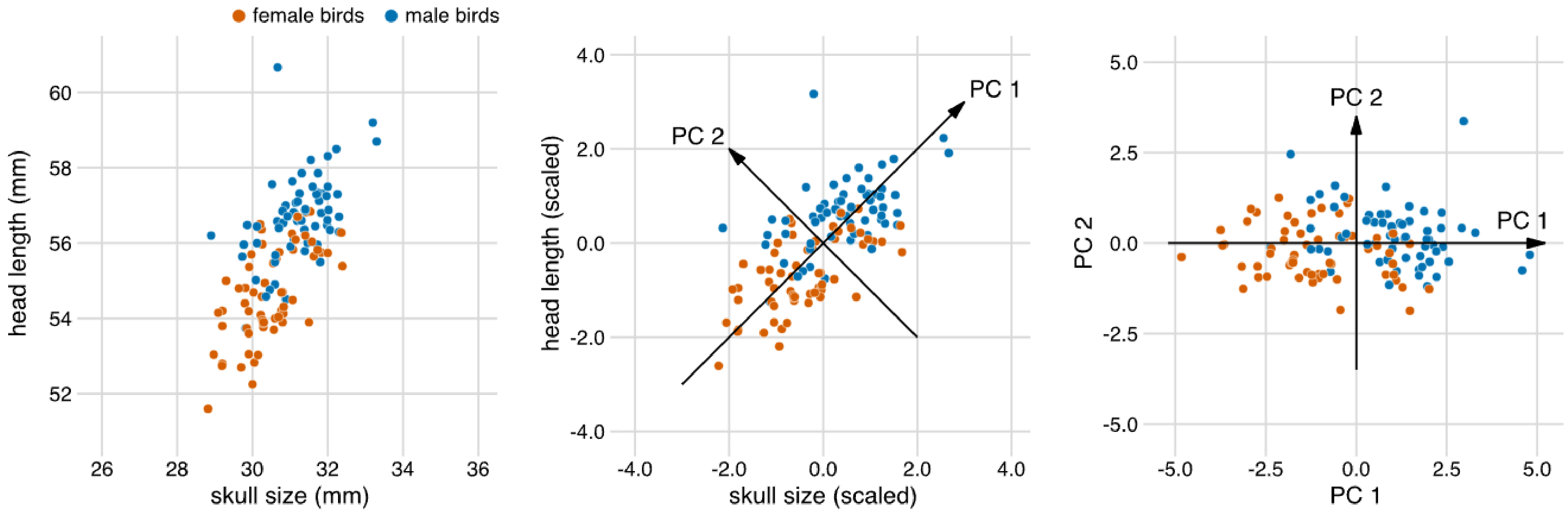



**Getting the old data back**

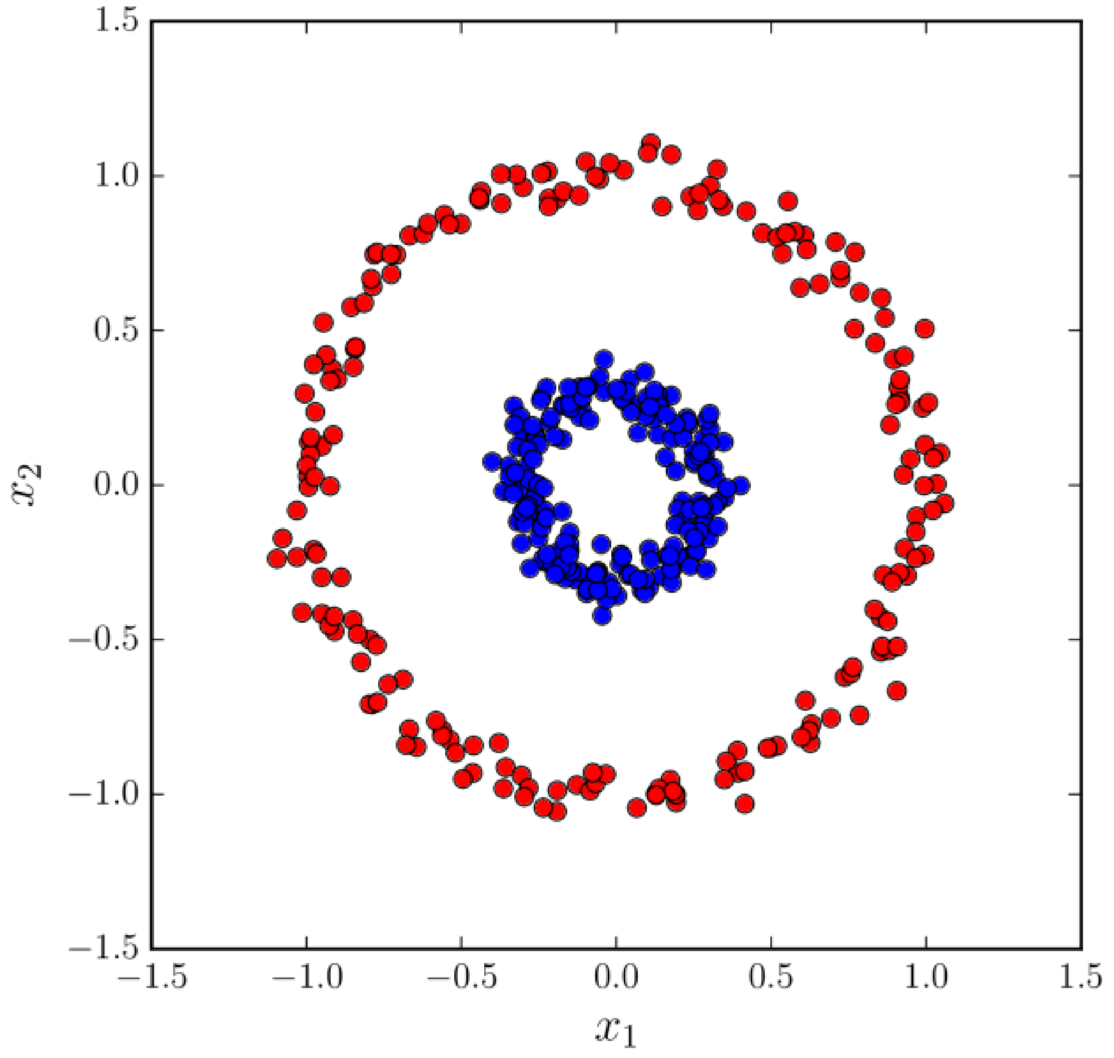
$$RowOriginalData = (RowFeatureVector^T \times FinalData) + OriginalMean$$

A tutorial on Principal Components Analysis by Lindsay I. Smith

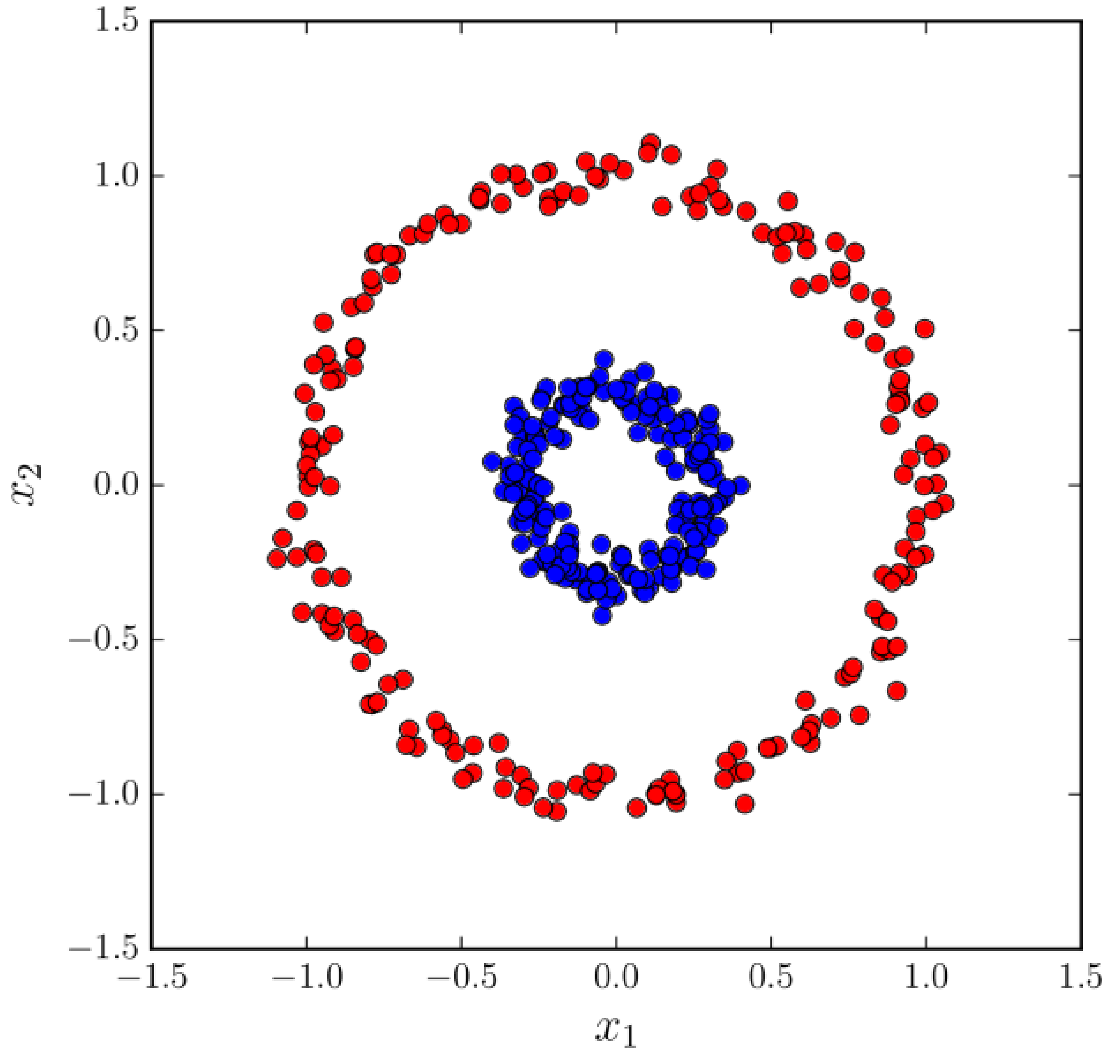
# Feature Extraction | Principal Components Analysis



# Feature Extraction | Principal Components Analysis



# Feature Extraction | Principal Components Analysis




# Feature Extraction

- Accuracy improvements.
- Overfitting risk reduction.
- Speed up in training.
- Improved Data Visualization.
- Increase in explainability of our model.

Many other techniques…

For a visual explanation of PCA, see https://setosa.io/ev/principal-component-analysis/