Principal Component Analysis

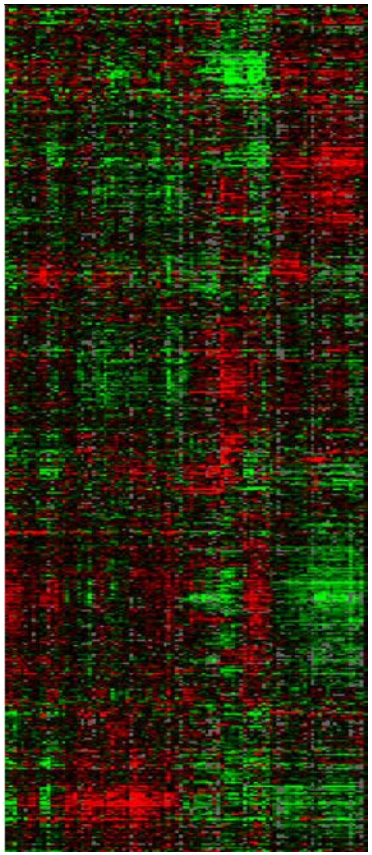
Chris Ding
Department of Computer Science and Engineering
University of Texas at Arlington

PCA is the procedure of finding intrinsic dimensions of the data

1. Data analysis
2. Data reduction
3. Data visualization

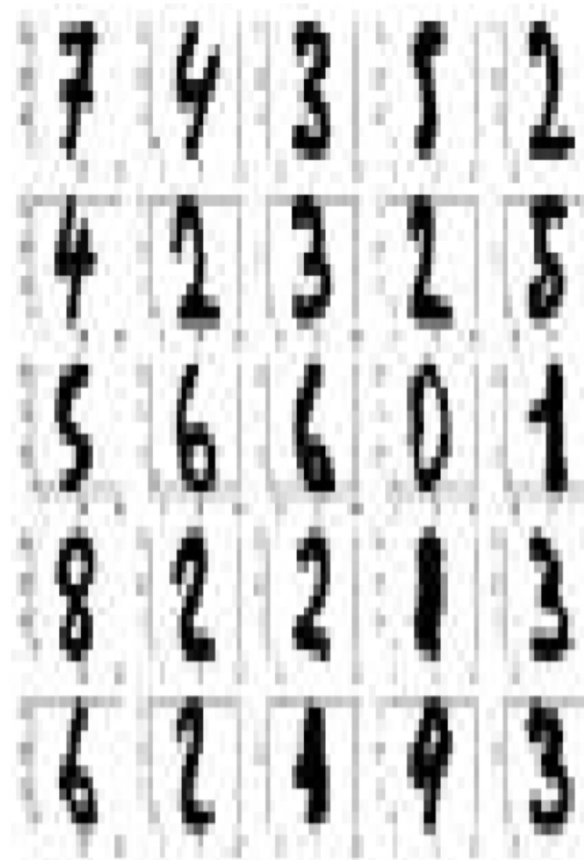
Represent high dimensional data in low-dim space

High-dimensional data

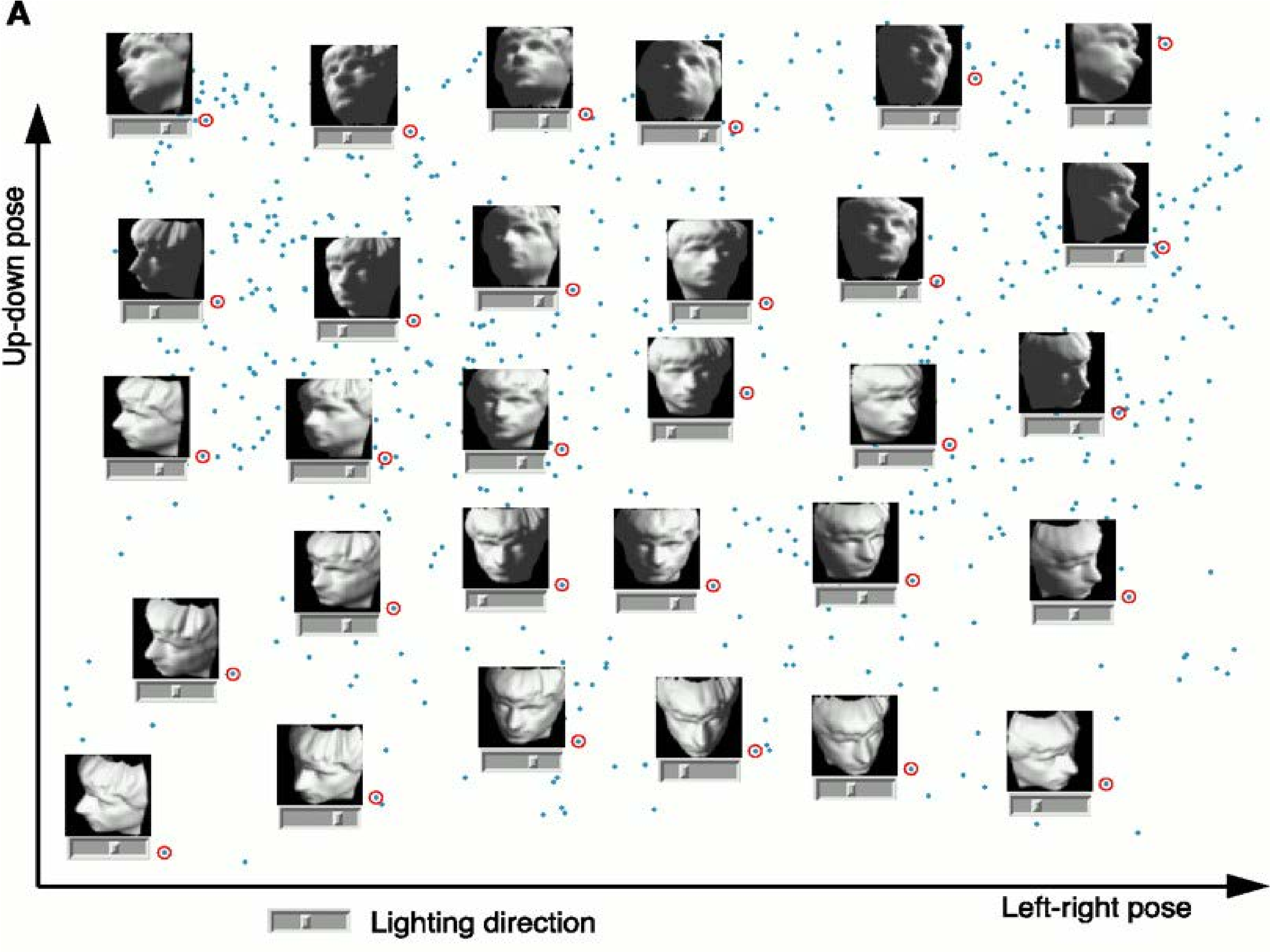


Gene expression

Face images

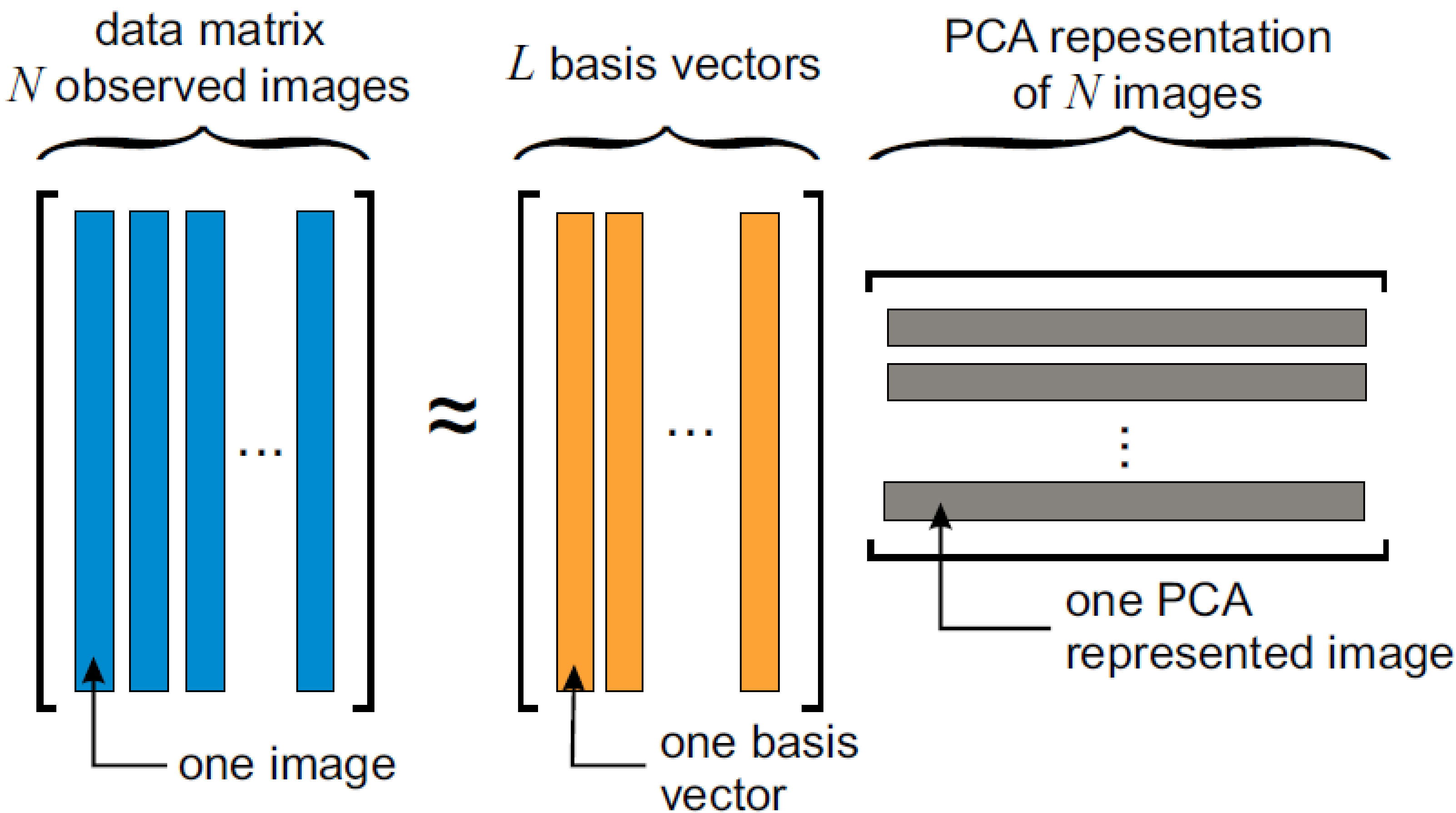


Handwritten digits


A
Up-down pose
Left-right pose
Lighting direction

Application of feature reduction

- Face recognition
- Handwritten digit recognition
- Text mining
- Image retrieval
- Microarray data analysis
- Protein classification


data matrix
N observed images
$\approx$
L basis vectors
PCA repesentation
of N images
one image
one basis
vector
one PCA
represented image

Use PCA to approximate an image (a data matrix)

112 x 92

original

PCA
k=10

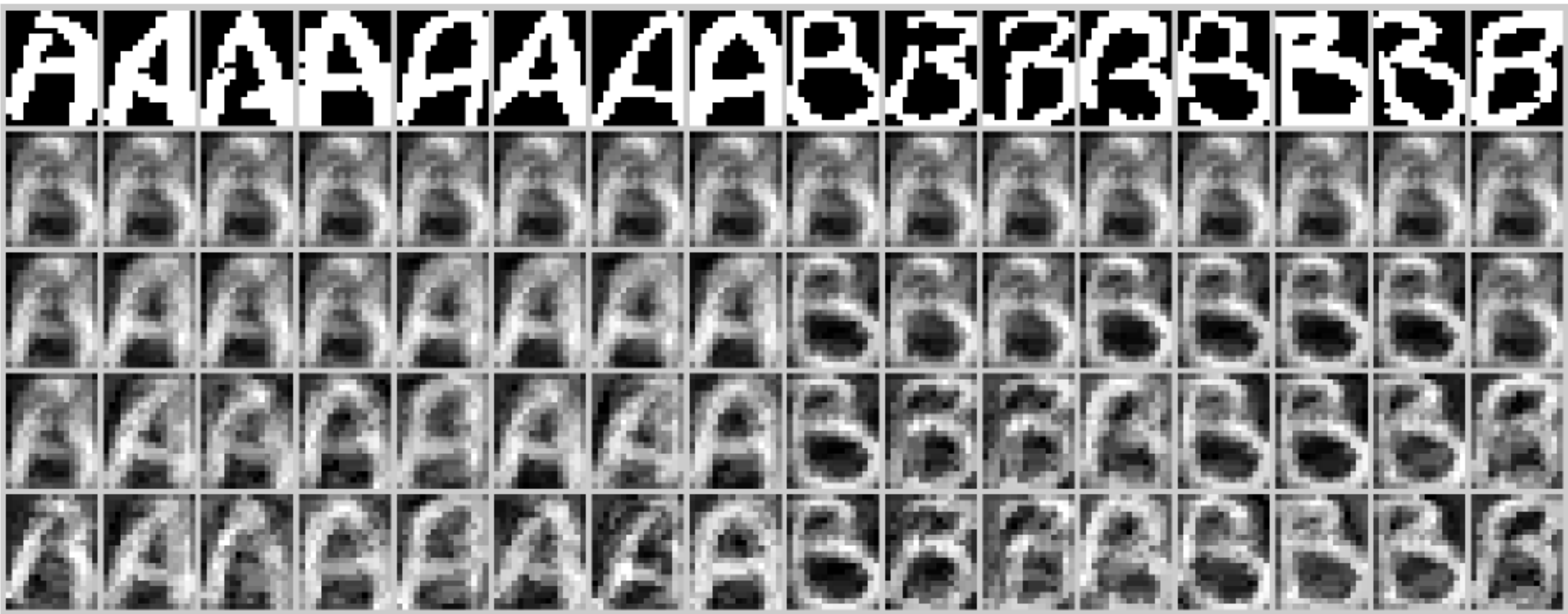
PCA
k=20

PCA
k=30

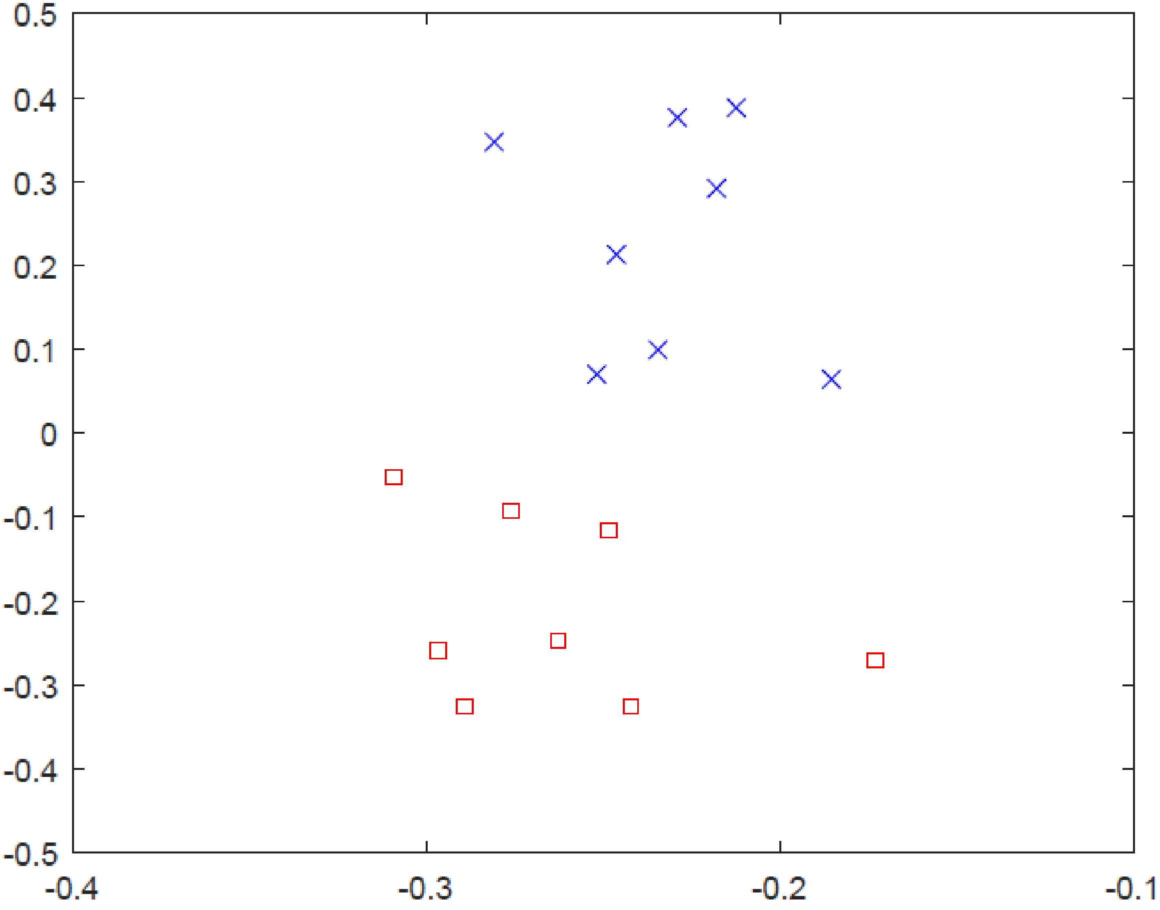
PCA
k=40

Use PCA to approximate a set of images

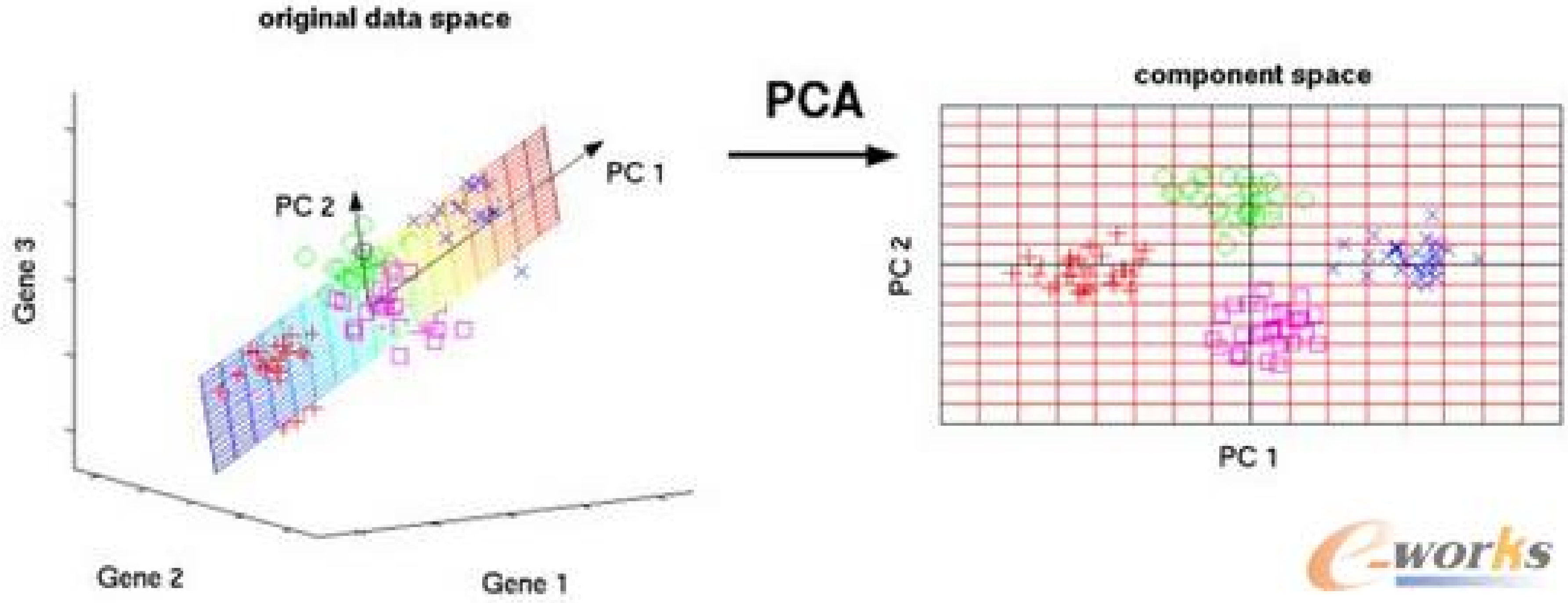
Use PCA to approximate a set of images



Display the characters in 2-dim space

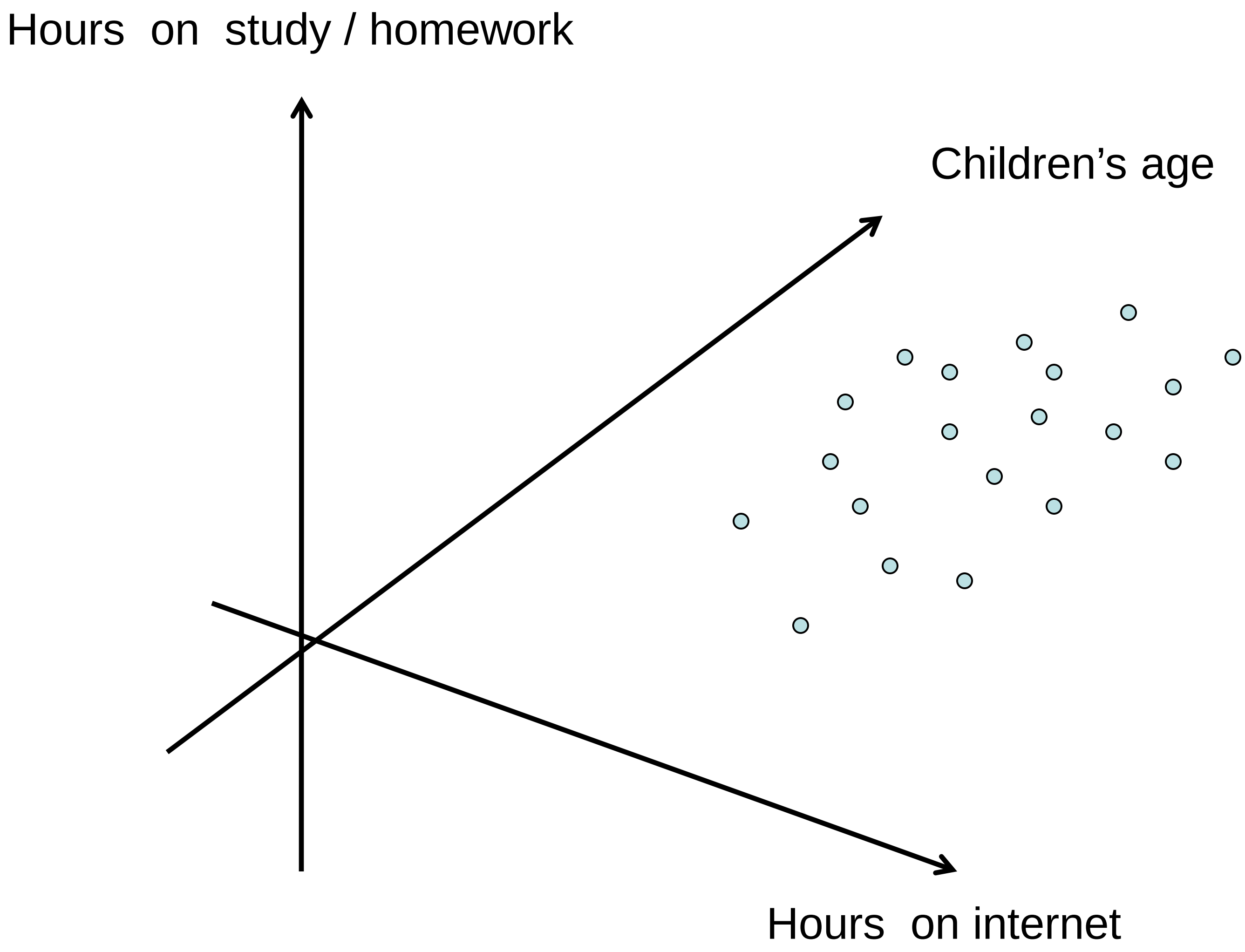


Application of feature reduction



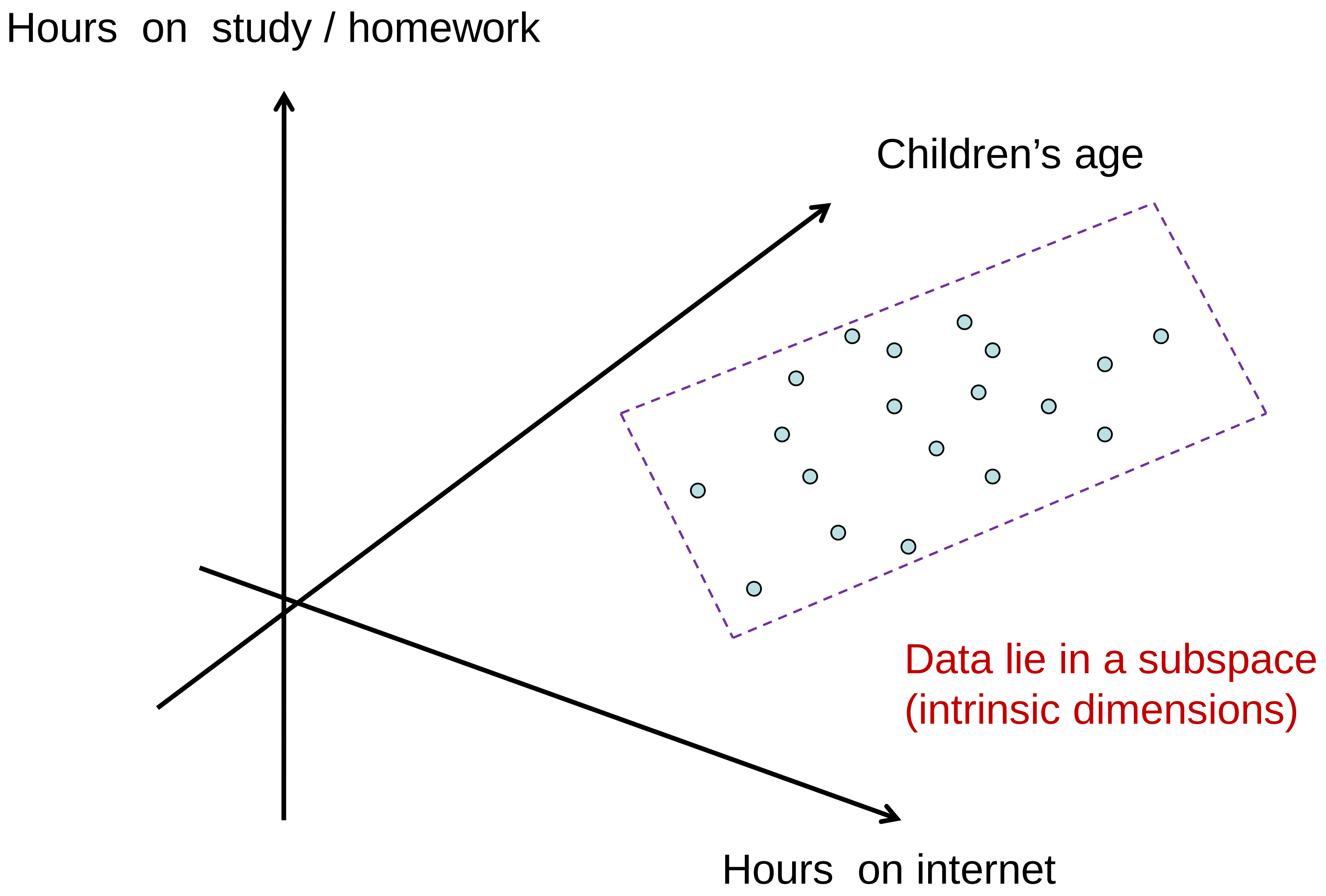
Intrinsic dimensions of the data

Samples of children: hours of study, hours on internet, vs their age



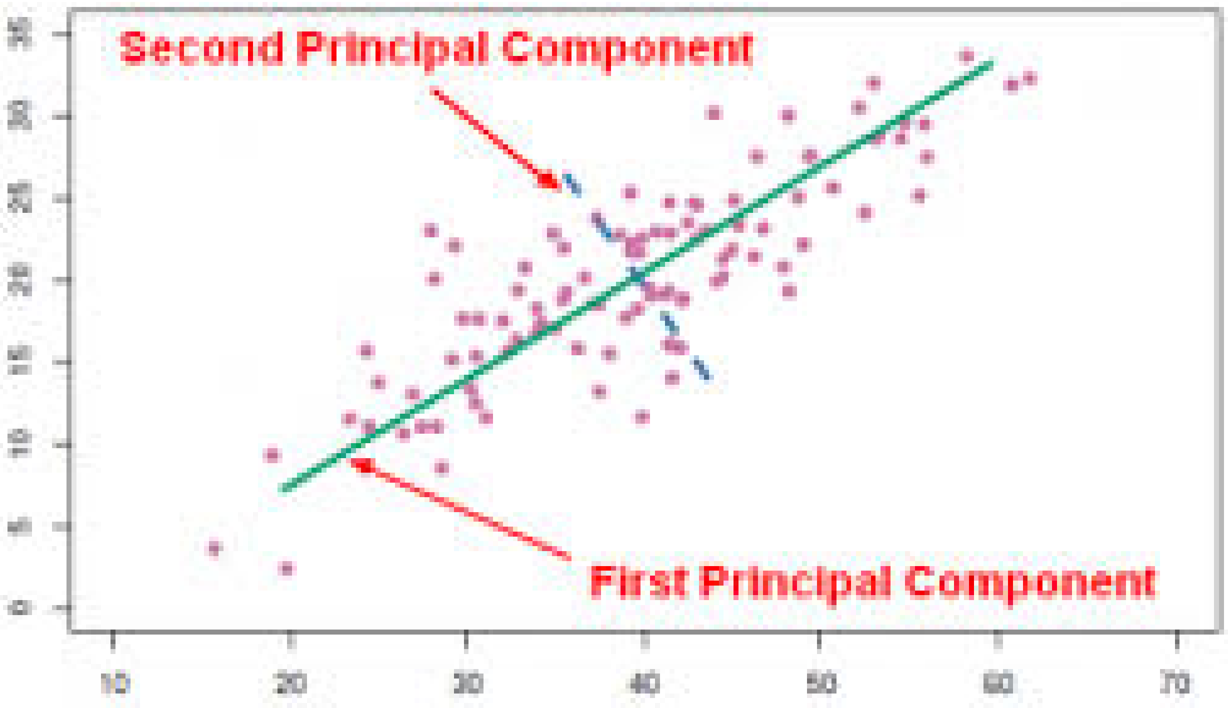
Intrinsic dimensions of the data

Samples of children: hours of study, hours on internet, vs their age

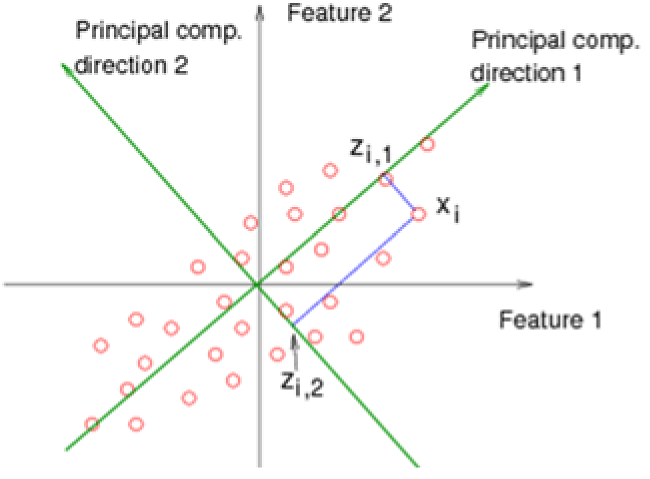


PCA is the procedure of finding intrinsic dimensions of the data

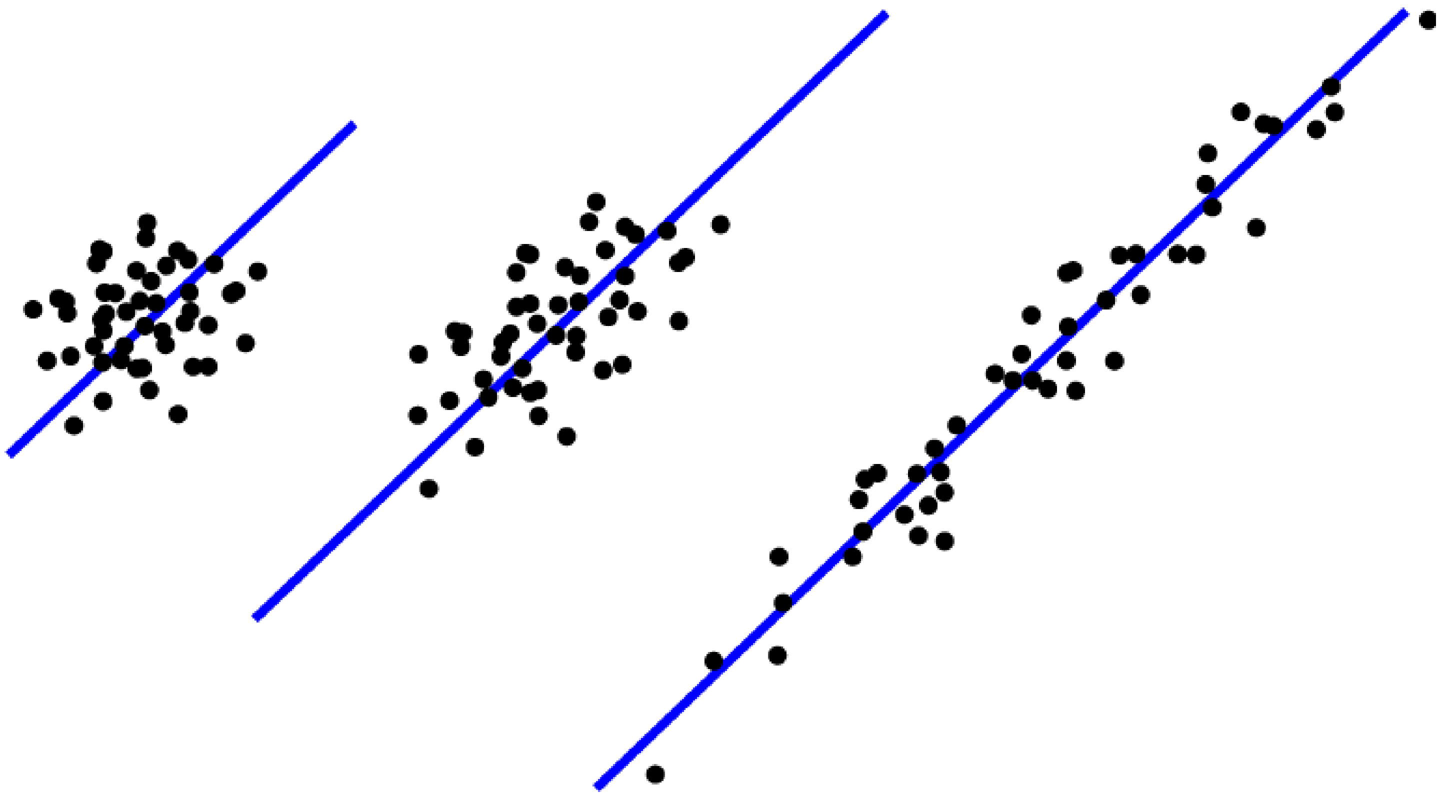
Find lines that best represent the data

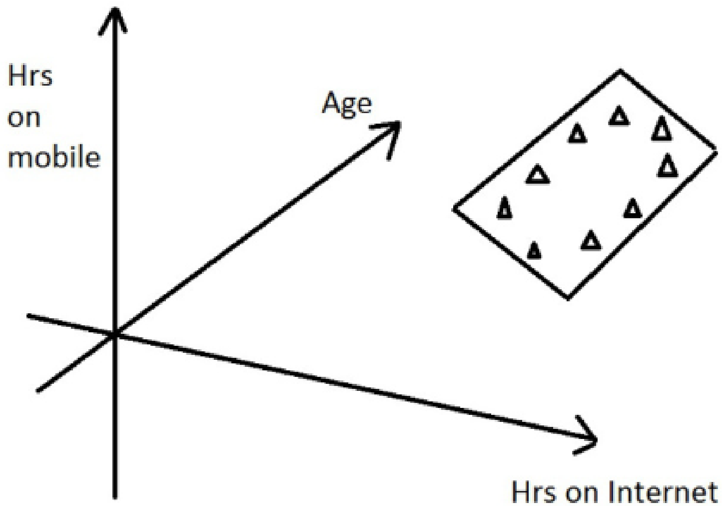


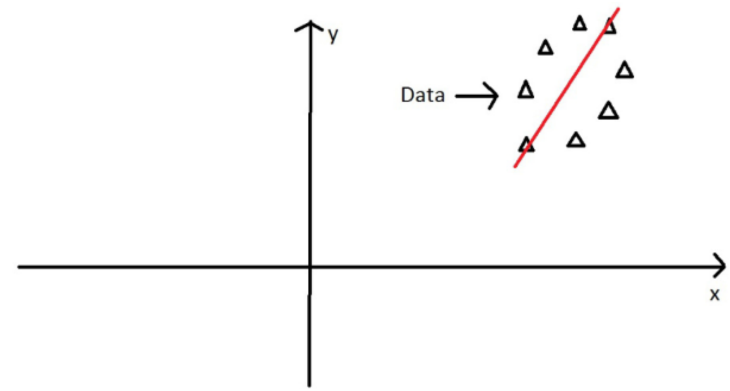
PCA is a rotation of space to proper directions (principal directions)

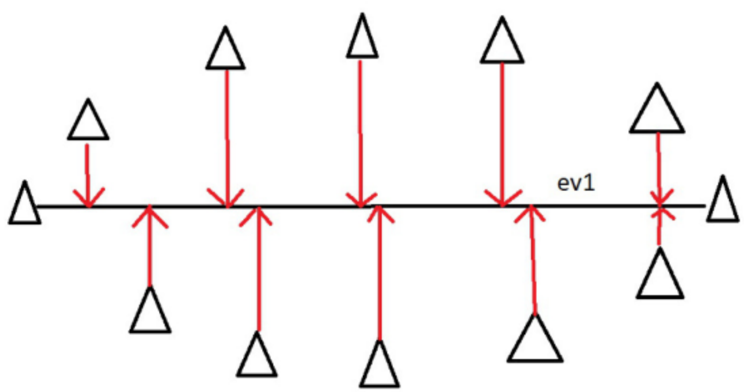


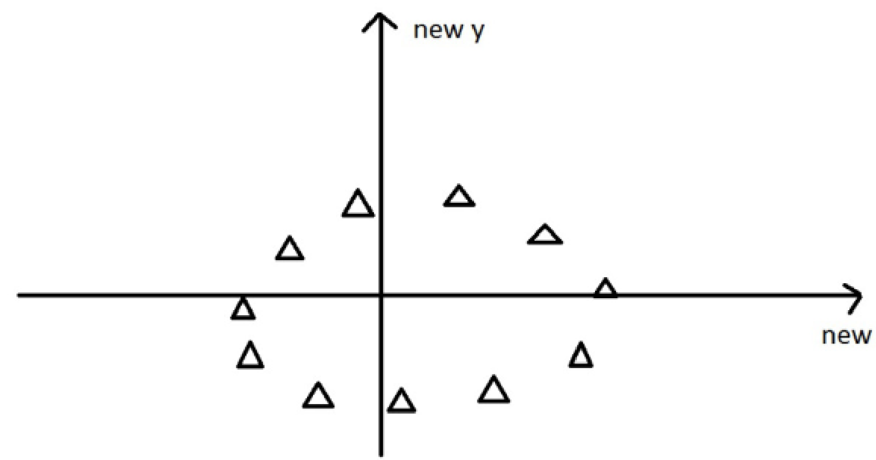
PCA represents data:
the close data to a linear subspace,
the more accurate representation

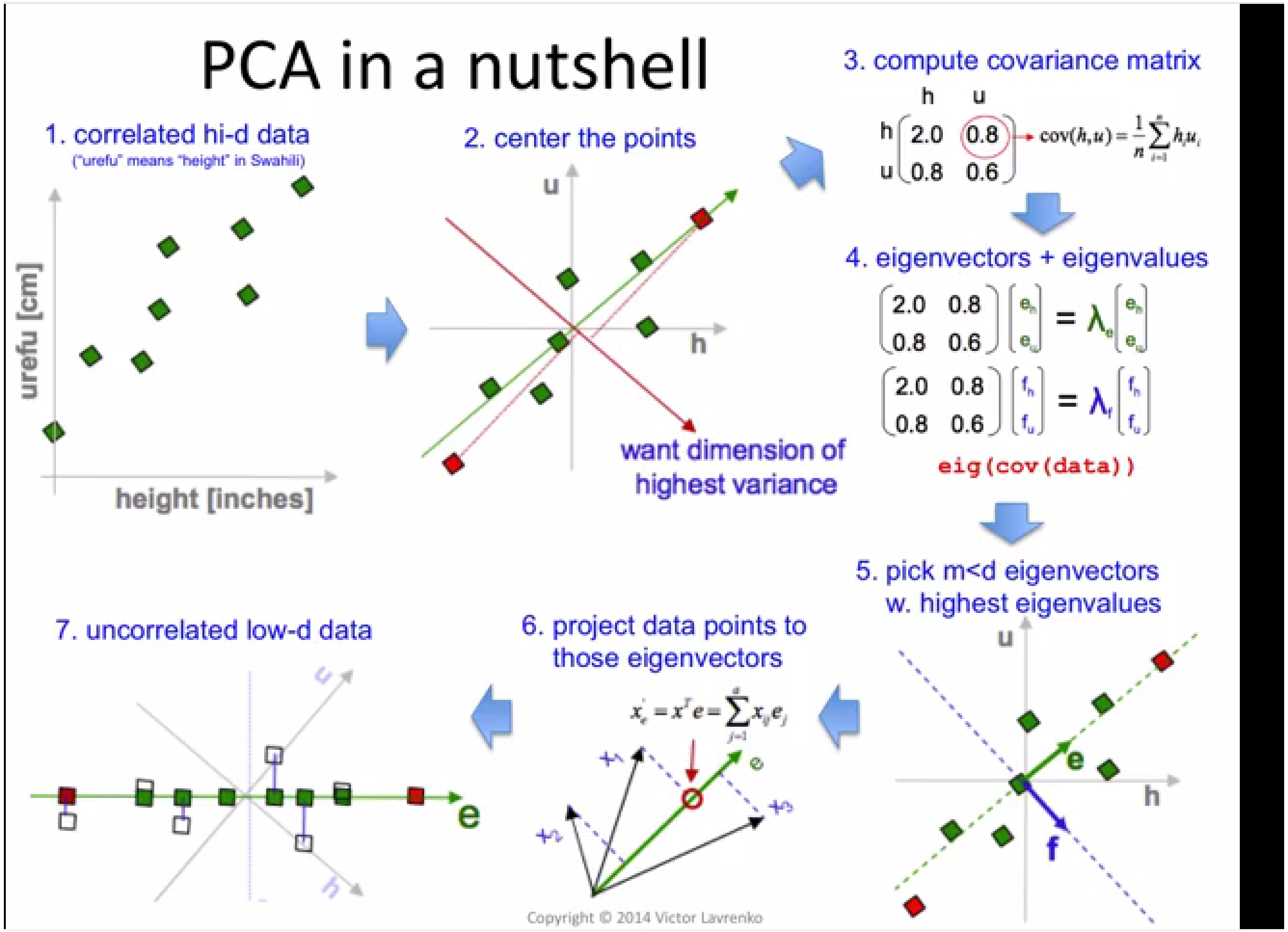



Hrs
on
mobile
Age
Hrs on Internet


y
Data
x

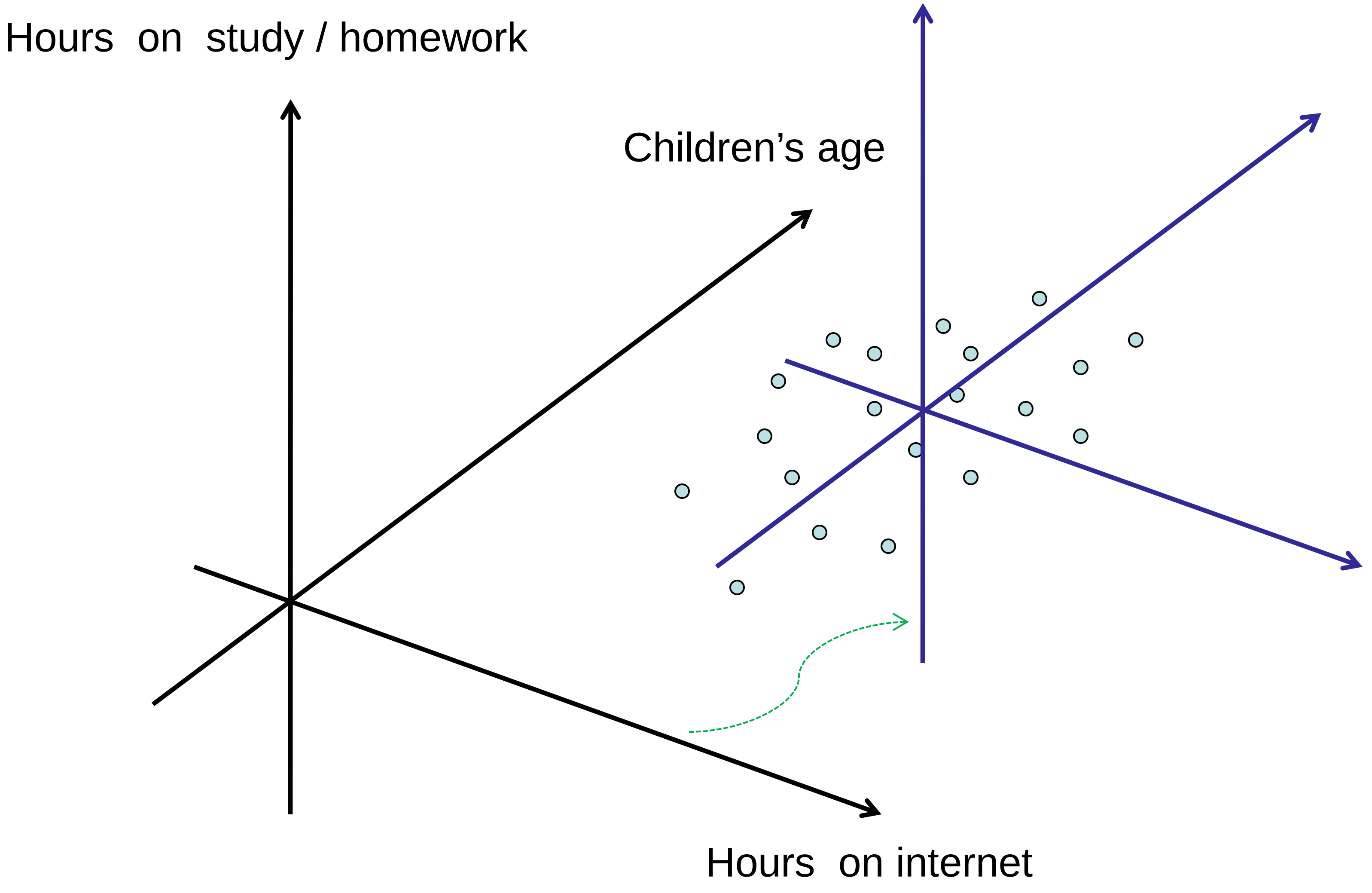

ev1


new y
new x

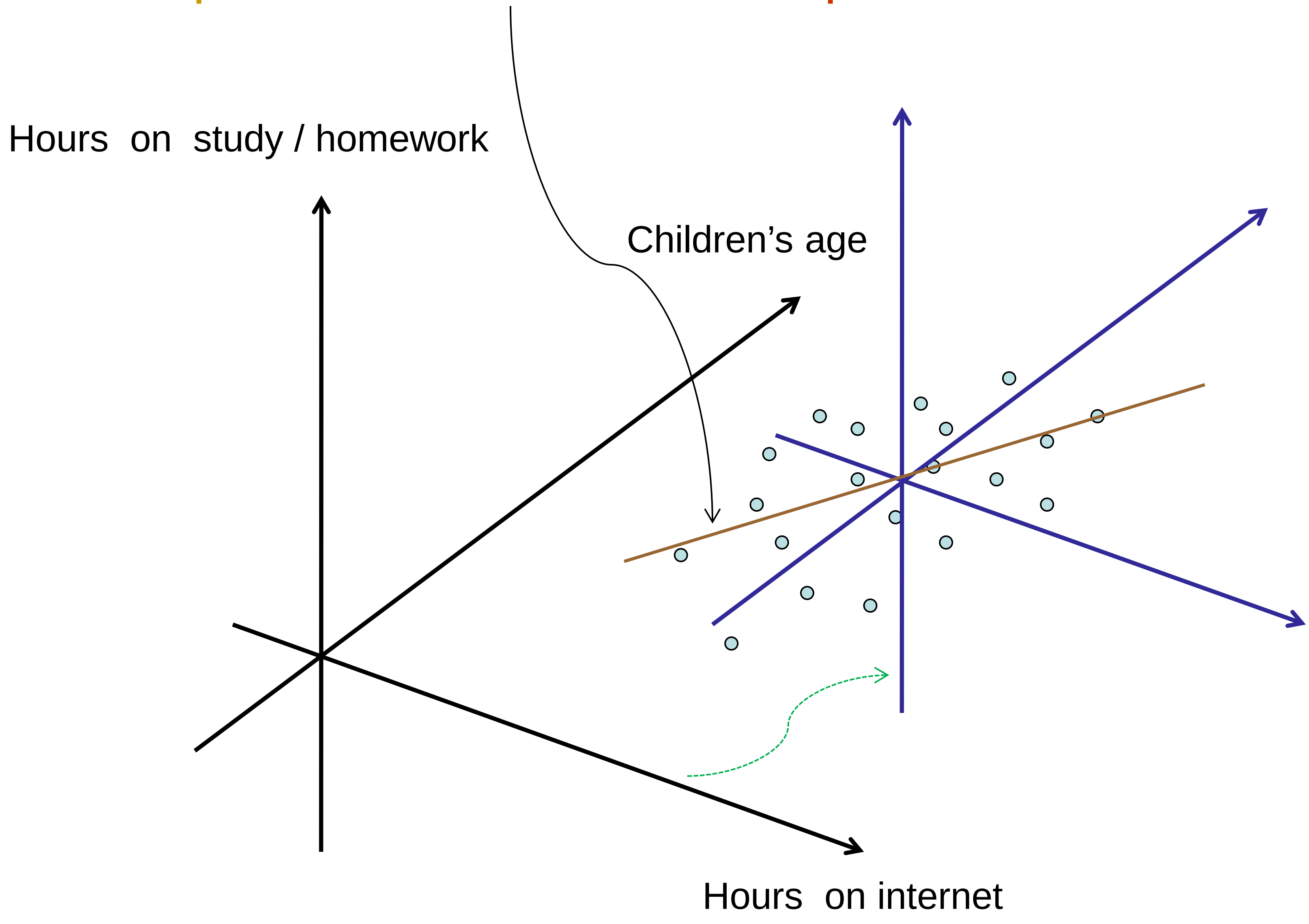

PCA in a nutshell
1. correlated hi-d data
("urefu" means "height" in Swahili)
urefu [cm]
height [inches]
2. center the points
u
h
want dimension of highest variance
3. compute covariance matrix
h u
h 2.0 0.8
u 0.8 0.6
$\mathrm{cov}(h,u) = \frac{1}{n}\sum_{i=1}^{n} h_i u_i$
4. eigenvectors + eigenvalues
$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}\begin{pmatrix} e_h \\ e_u \end{pmatrix} = \lambda_e \begin{pmatrix} e_h \\ e_u \end{pmatrix}$
$\begin{pmatrix} 2.0 & 0.8 \\ 0.8 & 0.6 \end{pmatrix}\begin{pmatrix} f_h \\ f_u \end{pmatrix} = \lambda_f \begin{pmatrix} f_h \\ f_u \end{pmatrix}$
eig(cov(data))
5. pick m<d eigenvectors w. highest eigenvalues
u
e
h
f
6. project data points to those eigenvectors
$x'_e = x^T e = \sum_{j=1}^{d} x_{ij} e_j$
e
7. uncorrelated low-d data
e
Copyright © 2014 Victor Lavrenko

PCA Step 0: move coordinate to data center

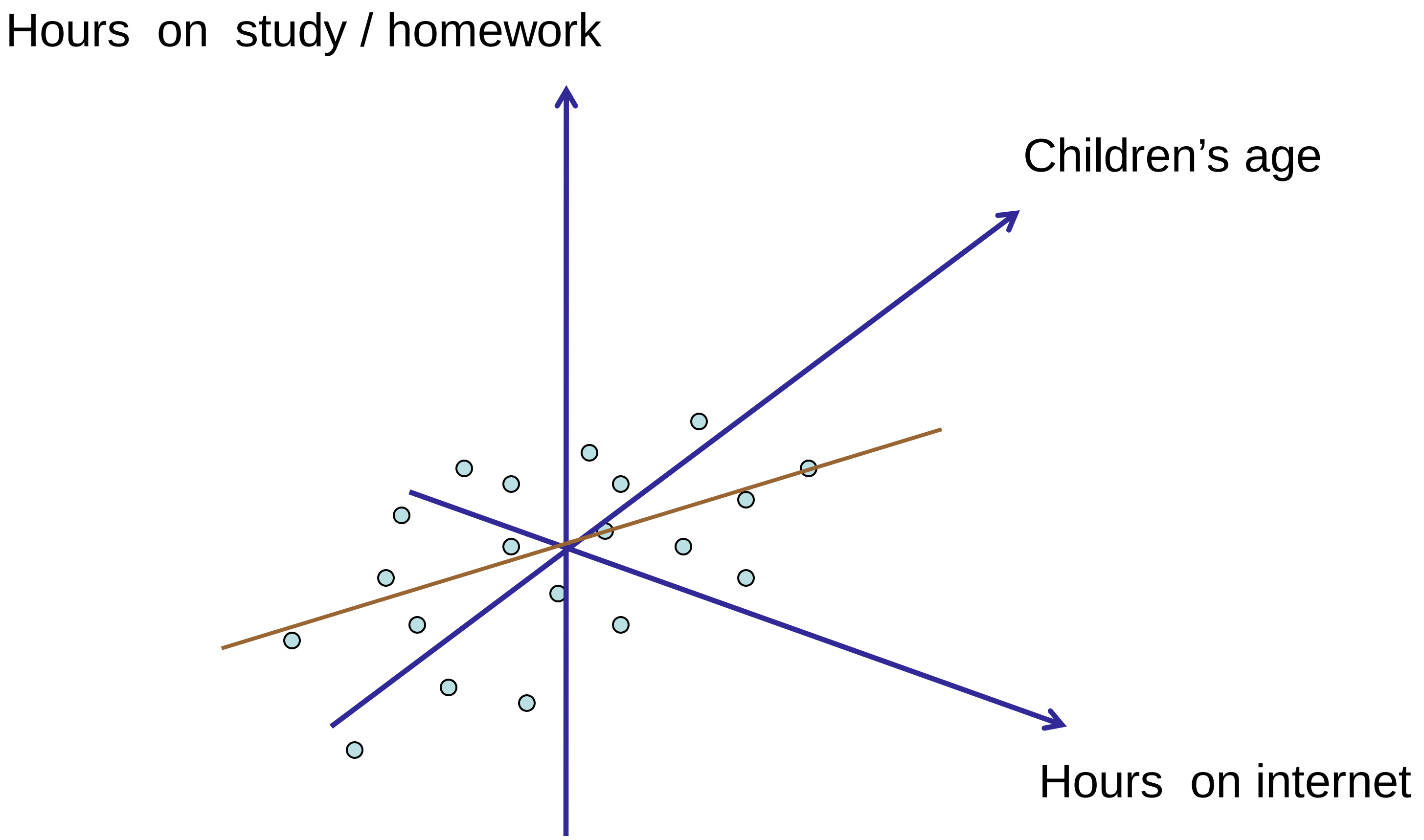
This is equivalent to Centering the data



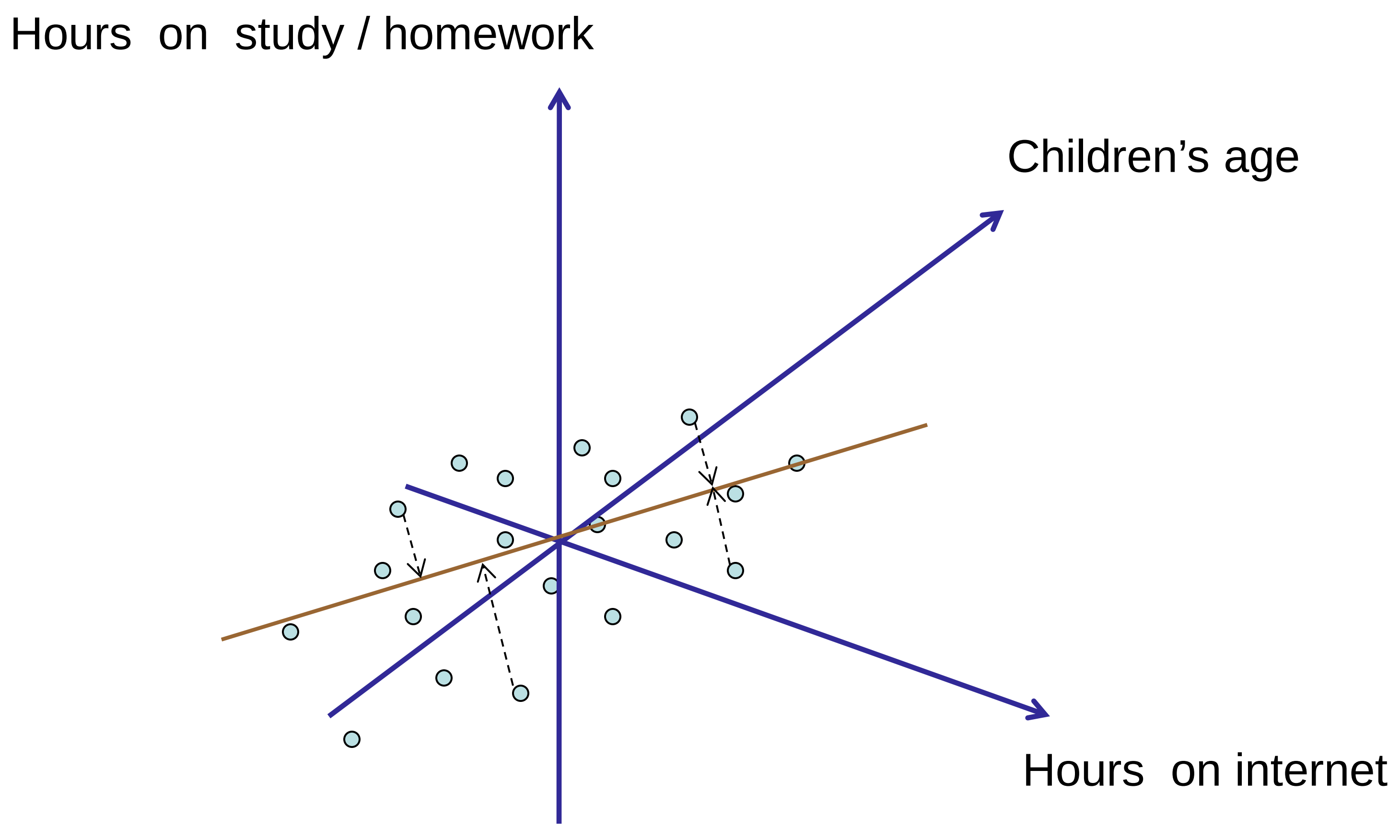
PCA Step 1: find a line that best represents the data



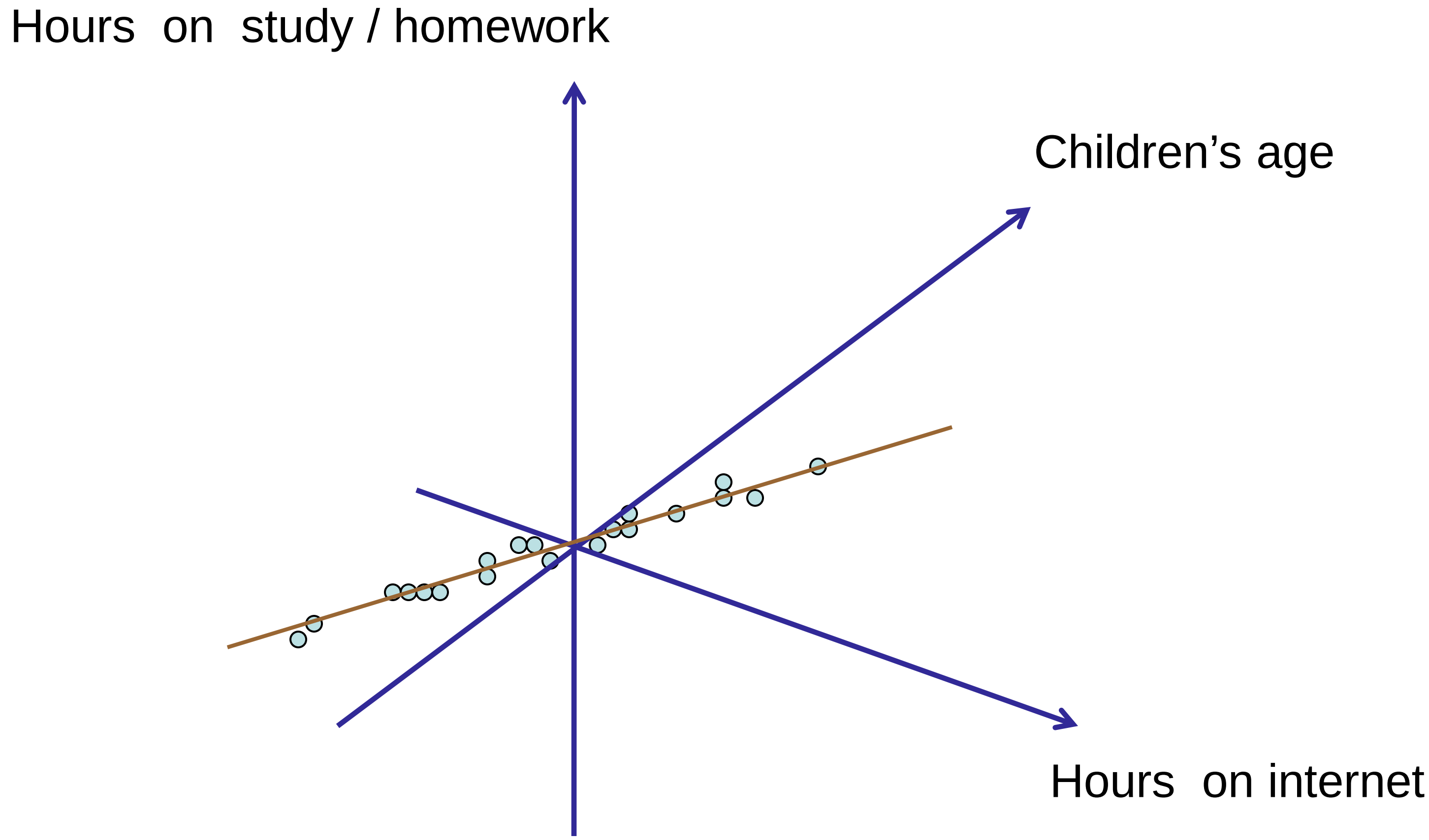
PCA Step 1: find a line that best represents the data



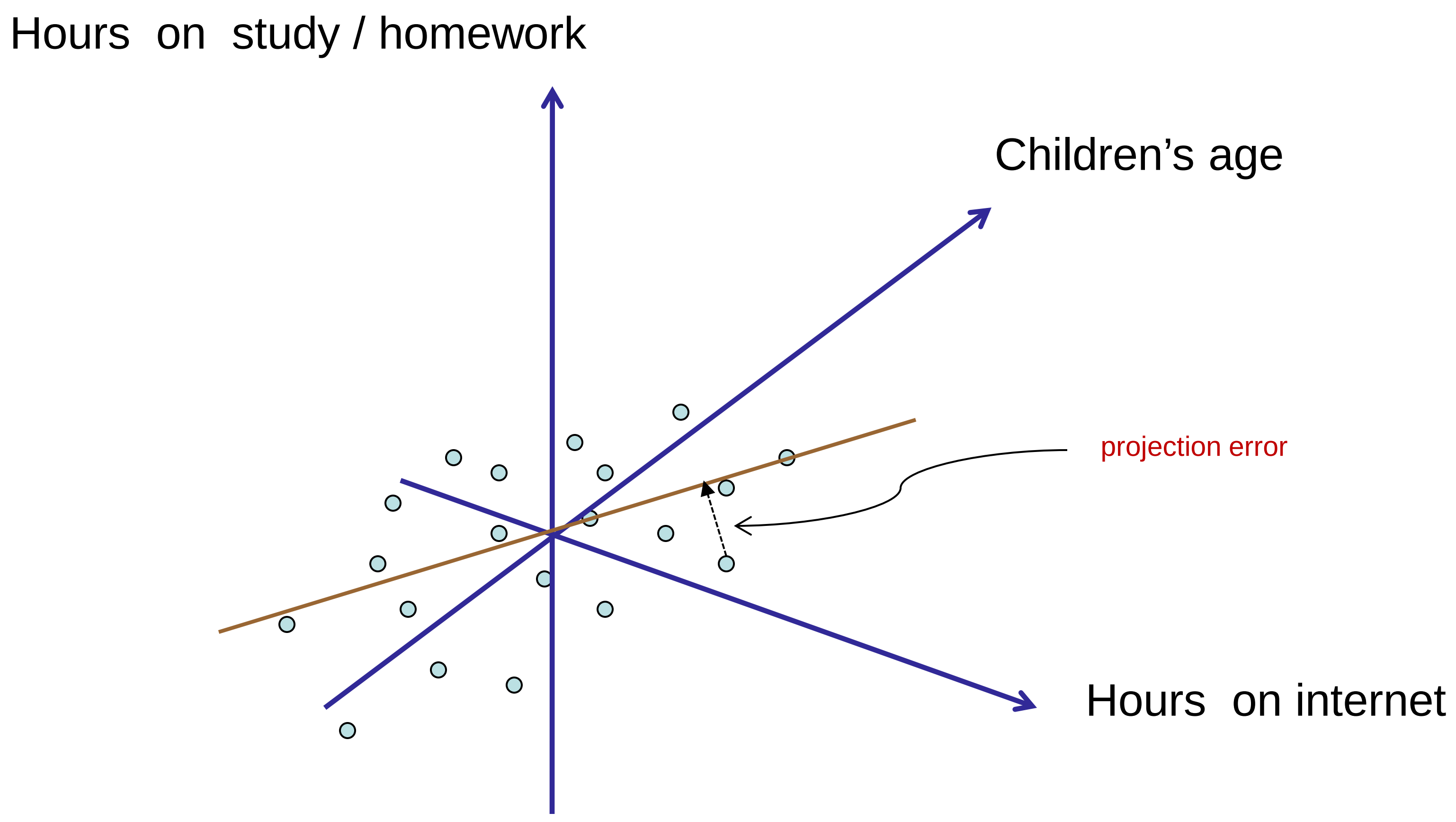
PCA Step 1: find a line that best represents the data



PCA Step 1: find a line that best represents the data

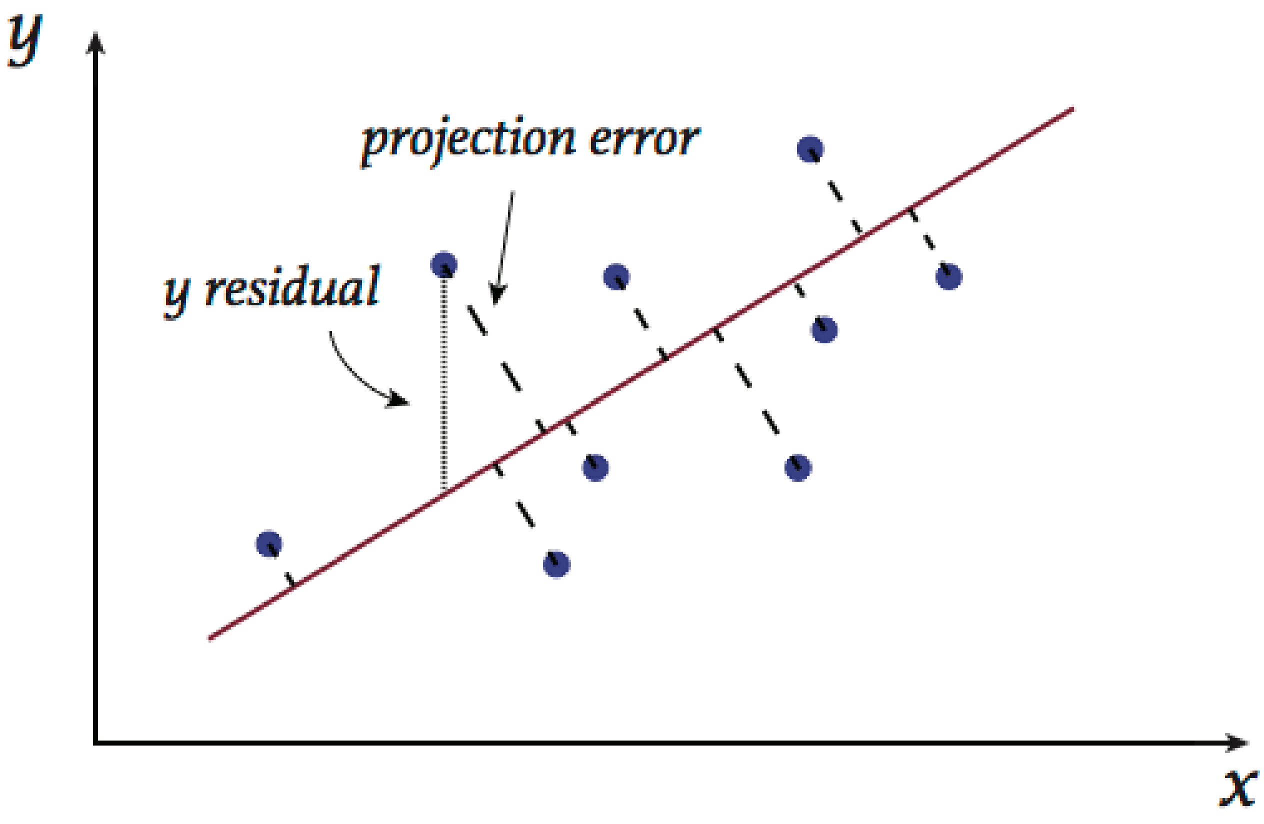


PCA Step 1: find a line that best represents the data

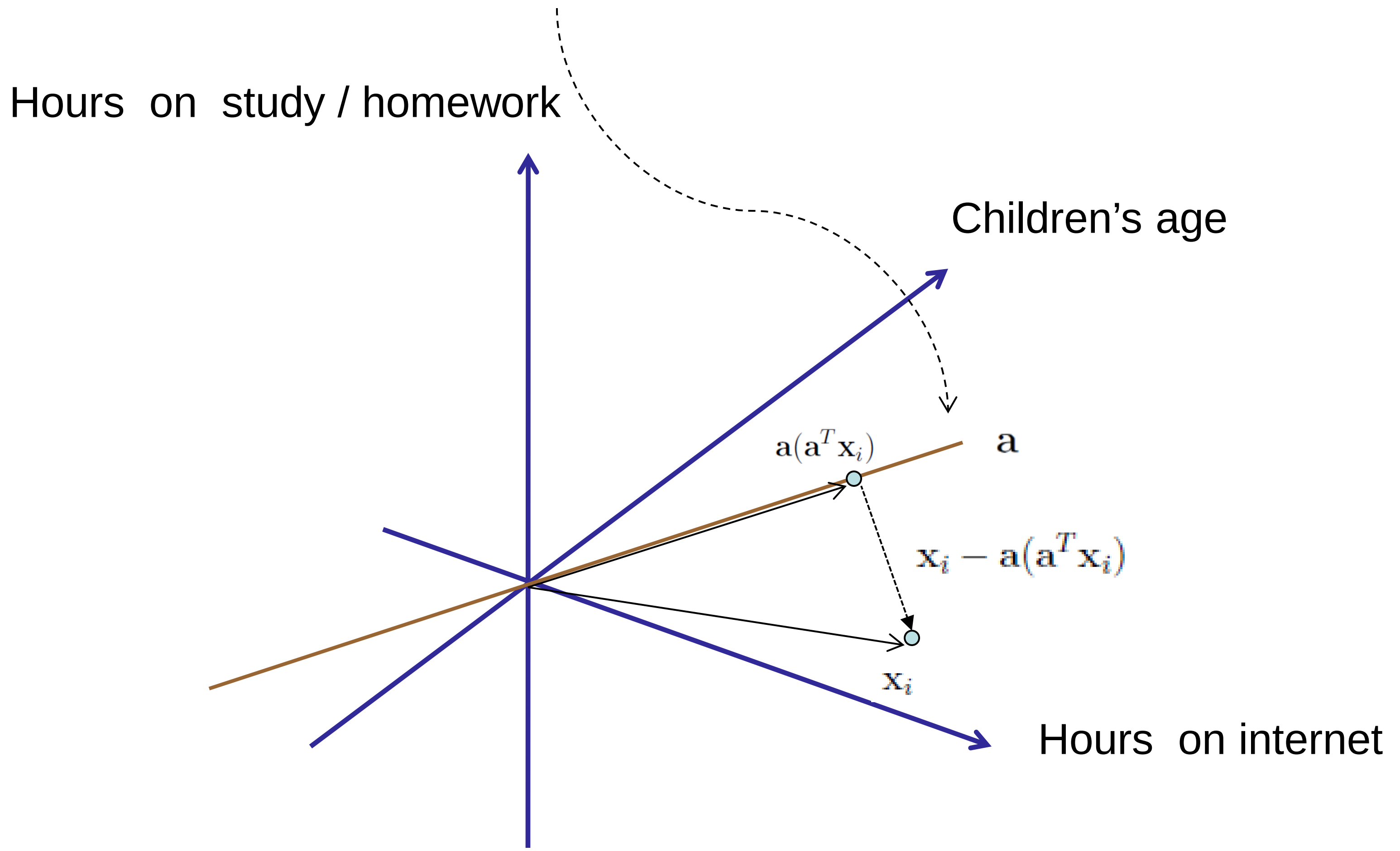


minimize sum of projection errors squared

PCA finds the best line by minimize the projection errors



PCA Step 1: find the line that best represents the data



minimize sum of projection errors squared

$$J(\mathbf{a}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{a}(\mathbf{a}^T\mathbf{x}_i)\|^2 = \|X - \mathbf{a}(\mathbf{a}^T X)\|_F^2$$

Note $\|A\|_F^2 = \mathrm{Tr} A^T A$, we have $J = \mathrm{Tr}(X^T X - 2X^T \mathbf{a}\mathbf{a}^T X + X^T \mathbf{a}\mathbf{a}^T \mathbf{a}\mathbf{a}^T X) = \mathrm{Tr}(X^T X - \mathbf{a}^T X X^T \mathbf{a})$. Therefore, the minimization of residual become

$$\max_{\mathbf{a}} \ \mathbf{a}^T X X^T \mathbf{a}, \ \ s.t. \ \mathbf{a}^T \mathbf{a} = 1 \tag{6}$$

The solution is given by the eigenvector of matrix XX^T associated wit the largest eigenvalue. Let the s.d.p. matrix XX^T has the following eigen-decomposition

$$XX^T = \sum_{k=1}^{r} \lambda_k \mathbf{u}_k \mathbf{u}_k^T = U \Lambda U^T, \ U = (\mathbf{u}_1 \cdots \mathbf{u}_r), \ \Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_r) \tag{7}$$

where the eigenvectors are ordered such as $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots$. and $r \leq (p, n)$ is the rank of X Therefore,

$$\mathbf{a}^* = \mathbf{u}_1,$$

This gives the 1st principal direction

Repeat this process to find 2nd, 3rd, … lines to best fit the remaining data, the results are given by u_2, u_3, …, uk.

In summary, the k-th order PCA of X is given by the k principal directions

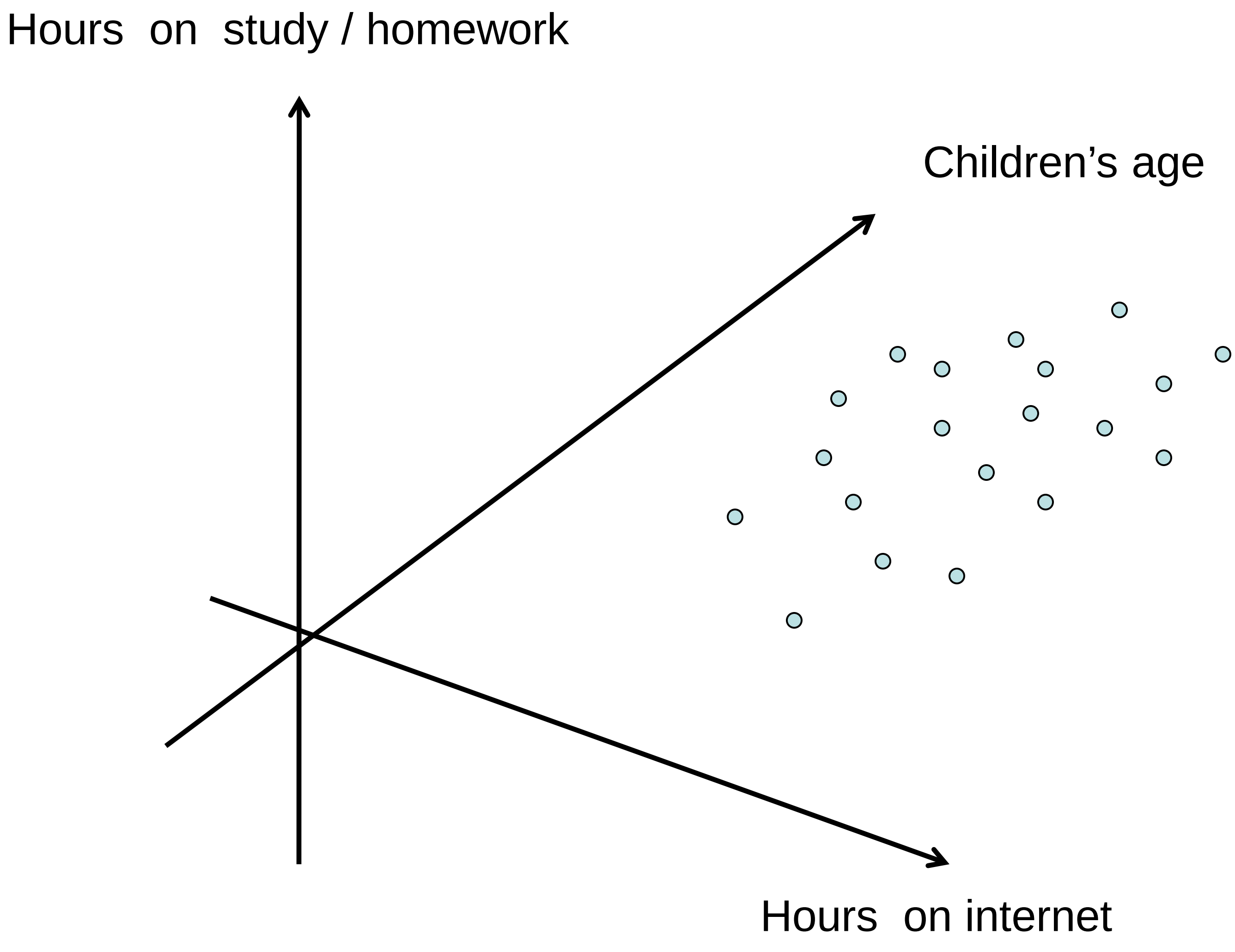
$$U_k = (\mathbf{u}_1 \cdots \mathbf{u}_k).$$

In the PCA subspace, each data is represented by k-dimensional vector

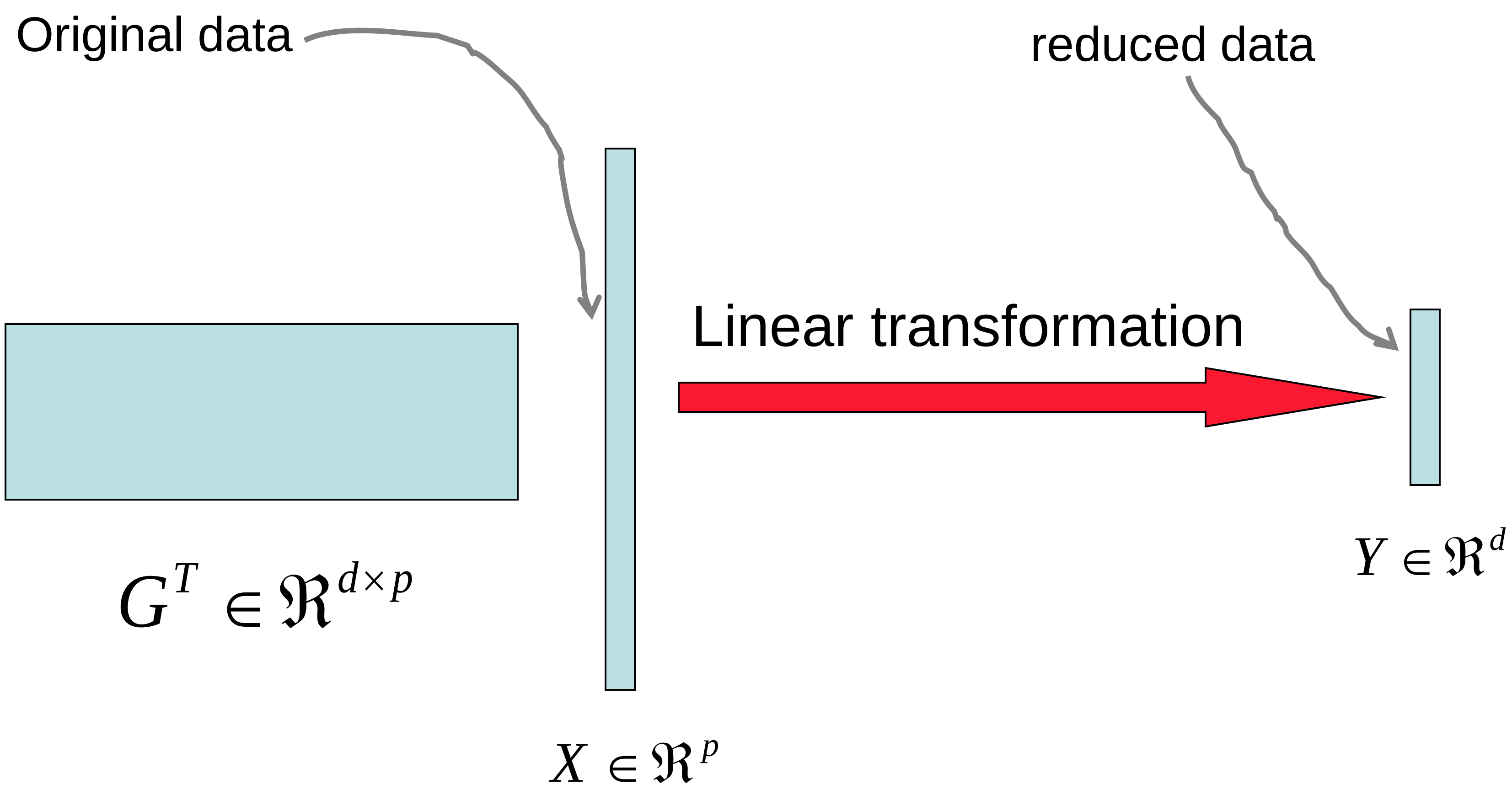
$$\mathbf{z}_i = U^T \mathbf{x}_i, \ i = 1 \cdots n, \ i = 1 \cdots n$$

Intrinsic dimensions of the data

Samples of children study, use internet vs their age



What is feature reduction?



$$G \in \Re^{p \times d} : X \to Y = G^T X \in \Re^d$$

Outline of lecture

- What is feature reduction?
- Why feature reduction?
- Feature reduction algorithms
- Principal Component Analysis
- Nonlinear PCA using Kernels

Why feature reduction?

- Most machine learning and data mining techniques may not be effective for high-dimensional data
  - Curse of Dimensionality
  - Query accuracy and efficiency degrade rapidly as the dimension increases.

- The intrinsic dimension may be small.
  - For example, the number of genes responsible for a certain type of disease may be small.

Why feature reduction?

- Visualization: projection of high-dimensional data onto 2D or 3D.

- Data compression: efficient storage and retrieval.

- Noise removal: positive effect on query accuracy.

Outline of lecture

- What is feature reduction?
- Why feature reduction?
- Feature reduction algorithms
- Principal Component Analysis
- Nonlinear PCA using Kernels

Feature reduction algorithms

- Unsupervised
  - Latent Semantic Indexing (LSI): truncated SVD
  - Independent Component Analysis (ICA)
  - Principal Component Analysis (PCA)
  - Canonical Correlation Analysis (CCA)
- Supervised
  - Linear Discriminant Analysis (LDA)
- Semi-supervised
  - Research topic

Outline of lecture

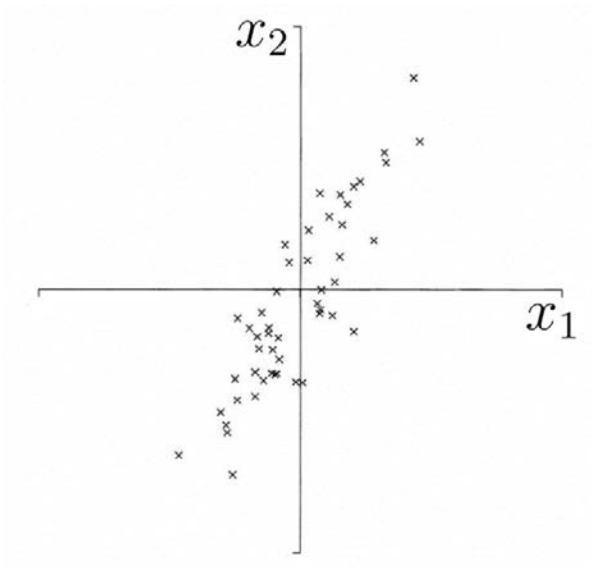
- What is feature reduction?
- Why feature reduction?
- Feature reduction algorithms
- Principal Component Analysis
- Nonlinear PCA using Kernels

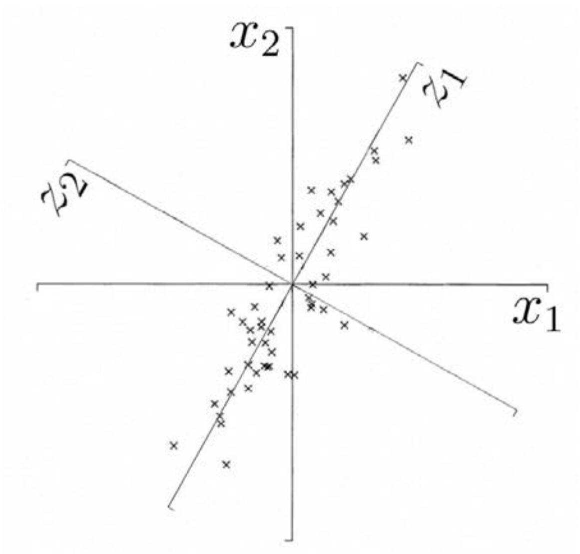
What is Principal Component Analysis?

- Principal component analysis (PCA)
  - Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables
  - Retains most of the sample's information.
  - Useful for the compression and classification of data.

- By information we mean the variation present in the sample, given by the correlations between the original variables.
  - The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

Geometric picture of principal components (PCs)





- the 1st PC Z_1 is a minimum distance fit to a line in X space
- the 2nd PC Z_2 is a minimum distance fit to a line in the plane perpendicular to the 1st PC

PCs are a series of linear least squares fits to a sample, each orthogonal to all the previous.

Algebraic definition of PCs

Given a sample of *n* observations on a vector of *p* variables

$$\{x_1, x_2, \cdots, x_n\} \in \Re^p$$

define the first principal component of the sample
by the linear transformation

$$z_1 = a_1^T x_j = \sum_{i=1}^{p} a_{i1} x_{ij}, \quad j = 1,2,\cdots,n.$$

where the vector

$$a_1 = (a_{11}, a_{21}, \cdots, a_{p1})$$

$$x_j = (x_{1j}, x_{2j}, \cdots, x_{pj})$$

is chosen such that $\mathrm{var}[z_1]$ is maximum.

Algebraic derivation of PCs

To find a_1 first note that

$$\text{var}[z_1] = E((z_1 - \overline{z_1})^2) = \frac{1}{n}\sum_{i=1}^{n}\left(a_1^T x_i - a_1^T \overline{x}\right)^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} a_1^T \left(x_i - \overline{x}\right)\left(x_i - \overline{x}\right)^T a_1 = a_1^T S a_1$$

where $$S = \frac{1}{n}\sum_{i=1}^{n}\left(x_i - \overline{x}\right)\left(x_i - \overline{x}\right)^T$$

is the covariance matrix. $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the mean.

In the following, we assume the Data is centered. $\overline{x} = 0$

Algebraic derivation of PCs

Assume $\bar{x} = 0$

Form the matrix: $X = [x_1, x_2, \cdots, x_n] \in \Re^{p \times n}$

then $S = \dfrac{1}{n} X X^T$

Obtain eigenvectors of S by computing the SVD of X:

$$X = U \Sigma V^T$$

Algebraic derivation of PCs

To find a_1 that maximizes $\text{var}[z_1]$ subject to $a_1^T a_1 = 1$

Let λ be a Lagrange multiplier

$$L = a_1^T S a_1 - \lambda(a_1^T a_1 - 1)$$

$$\frac{\partial}{\partial a_1} L = S a_1 - \lambda a_1 = 0$$

$$\Rightarrow (S - \lambda I_p) a_1 = 0$$

therefore a_1 is an eigenvector of S

corresponding to the largest eigenvalue $\lambda = \lambda_1$.

Algebraic derivation of PCs

To find the next coefficient vector a_2 maximizing $\mathrm{var}[z_2]$

subject to $\mathrm{cov}[z_2, z_1] = 0$

uncorrelated

and to $a_2^T a_2 = 1$

First note that $\mathrm{cov}[z_2, z_1] = a_1^T S a_2 = \lambda_1 a_1^T a_2$

then let λ and φ be Lagrange multipliers, and maximize

$$L = a_2^T S a_2 - \lambda(a_2^T a_2 - 1) - \phi a_2^T a_1$$

Algebraic derivation of PCs

$$L = a_2^T S a_2 - \lambda(a_2^T a_2 - 1) - \phi a_2^T a_1$$

$$\frac{\partial}{\partial a_2} L = S a_2 - \lambda a_2 - \phi a_1 = 0 \Rightarrow \phi = 0$$

$$S a_2 = \lambda a_2 \quad \text{and} \quad \lambda = a_2^T S a_2$$

Algebraic derivation of PCs

We find that a_2 is also an eigenvector of S

whose eigenvalue $\lambda = \lambda_2$ is the second largest.

In general

$$\mathrm{var}[z_k] = a_k^T S a_k = \lambda_k$$

- The k^{th} largest eigenvalue of S is the variance of the k^{th} PC.
- The k^{th} PC z_k retains the k^{th} greatest fraction of the variation in the sample.

Algebraic derivation of PCs

- Main steps for computing PCs
  - Form the covariance matrix S.
  - Compute its eigenvectors: $\{a_i\}_{i=1}^{p}$
  - Use the first d eigenvectors $\{a_i\}_{i=1}^{d}$ to form the d PCs.
  - The transformation G is given by

$$G \leftarrow [a_1, a_2, \cdots, a_d]$$

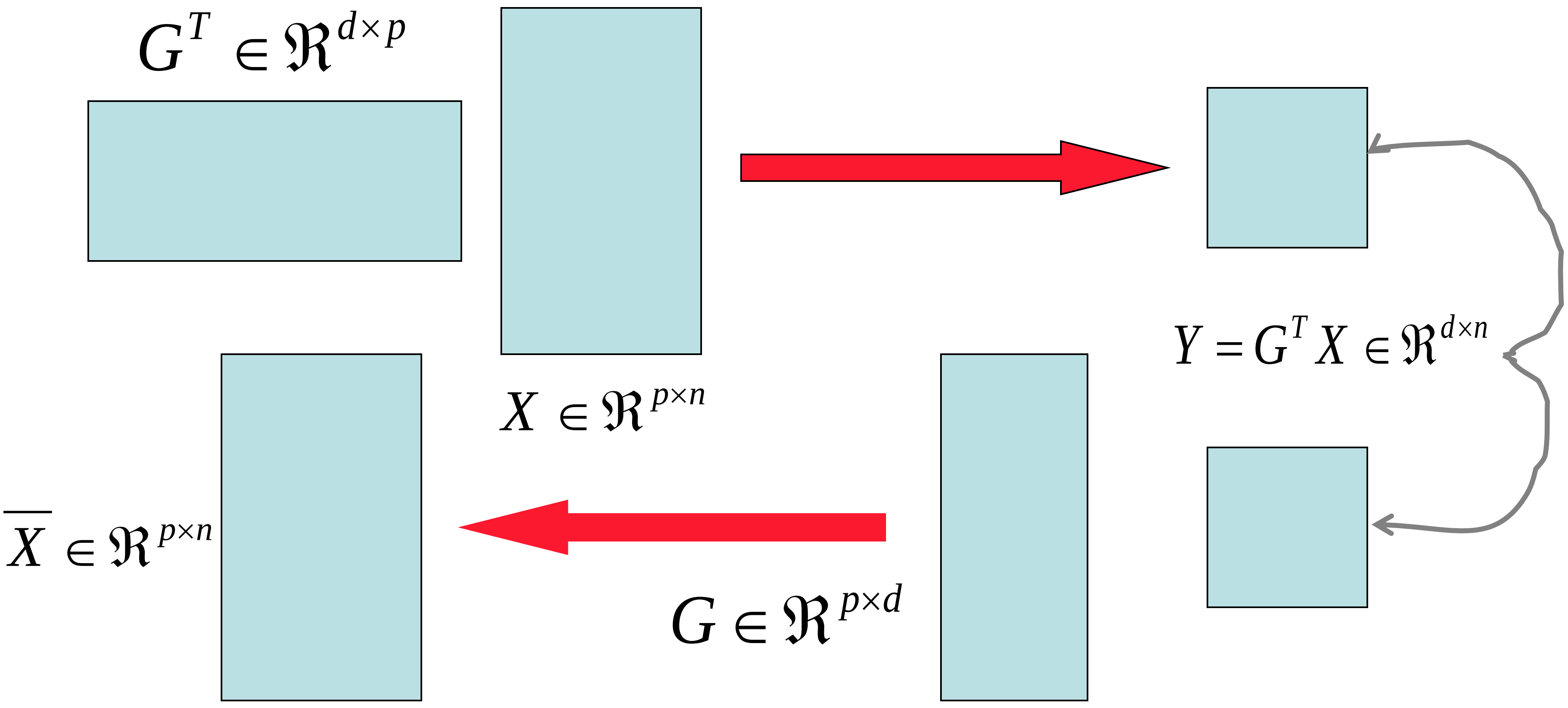
A test point $\boxed{x \in \Re^p \rightarrow G^T x \in \Re^d.}$

Optimality property of PCA

Reconstruction

Dimension reduction $X \in \Re^{p\times n} \rightarrow G^T X \in \Re^{d\times n}$

Original data $G^T X \in \Re^{d\times n} \rightarrow \boxed{\overline{X} = G(G^T X) \in \Re^{p\times n}}$



Optimality property of PCA

Main theoretical result:

The matrix G consisting of the first d eigenvectors of the covariance matrix S solves the following min problem:

$$\boxed{\min_{G \in \Re^{p \times d}} \left\| X - G(G^T X) \right\|_F^2 \text{ subject to } G^T G = I_d}$$

$$\left\| X - \overline{X} \right\|_F^2$$

reconstruction error

PCA projection minimizes the reconstruction error among all linear projections of size d.

Applications of PCA

- *Eigenfaces for recognition*. Turk and Pentland. 1991.
- *Principal Component Analysis for clustering gene expression data*. Yeung and Ruzzo. 2001.
- *Probabilistic Disease Classification of Expression-Dependent Proteomic Data from Mass Spectrometry of Human Serum.* Lilien. 2003.

PCA for image compression

d=1

d=2

d=4

d=8

d=16

d=32

d=64

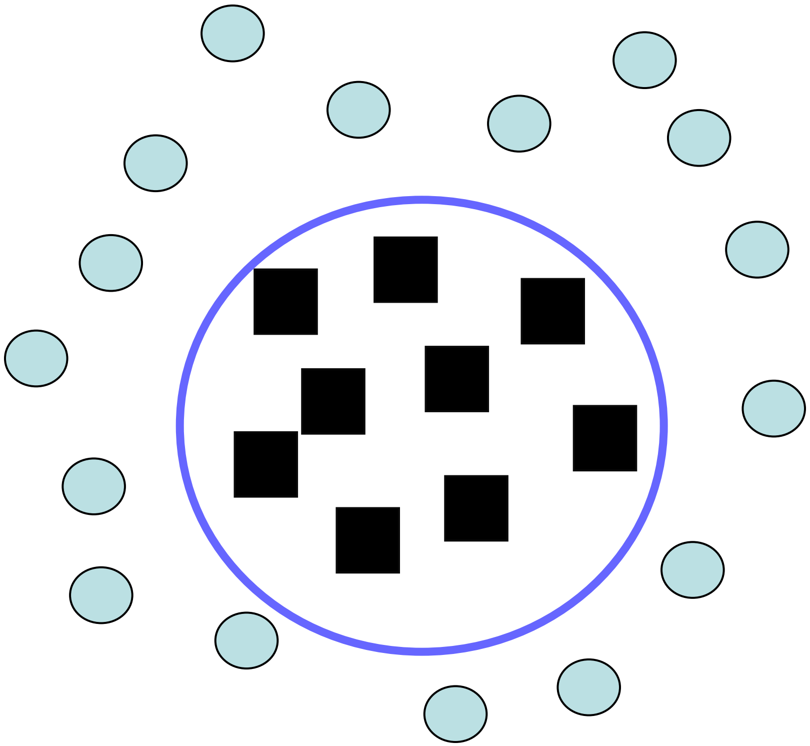
d=100

Original Image

Outline of lecture

- What is feature reduction?
- Why feature reduction?
- Feature reduction algorithms
- Principal Component Analysis
- Nonlinear PCA using Kernels

Motivation



Linear projections will not detect the pattern.

Nonlinear PCA using Kernels

- Traditional PCA applies linear transformation
  - May not be effective for nonlinear data

- Solution: apply nonlinear transformation to potentially very high-dimensional space.

$$\phi : x \rightarrow \phi(x)$$

- Computational efficiency: apply the kernel trick.
  - Require PCA can be rewritten in terms of dot product.

$$K(x_i, x_j) = \phi(x_i) \bullet \phi(x_j)$$

More on kernels later

Nonlinear PCA using Kernels

Rewrite PCA in terms of dot product

Assume the data has been centered, i.e., $\sum_i x_i = 0.$

The covariance matrix S can be written as $S = \frac{1}{n}\sum_i x_i x_i^T$

Let v be The eigenvector of S corresponding to nonzero eigenvalue

$$\boxed{Sv = \frac{1}{n}\sum_i x_i x_i^T v = \lambda v \Rightarrow v = \frac{1}{n\lambda}\sum_i (x_i^T v) x_i}$$

Eigenvectors of S lie in the space spanned by all data points.

Nonlinear PCA using Kernels

$$Sv = \frac{1}{n}\sum_i x_i x_i^T v = \lambda v \Rightarrow v = \frac{1}{n\lambda}\sum_i (x_i^T v) x_i$$

The covariance matrix can be written in matrix form:

$$S = \frac{1}{n} X X^T, \text{ where } \mathrm{X} = [\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_\mathrm{n}].$$

$$v = \sum_i \alpha_i x_i = X\alpha \qquad Sv = \frac{1}{n} X X^T X\alpha = \lambda X\alpha$$

$$\frac{1}{n}(X^T X)(X^T X)\alpha = \lambda (X^T X)\alpha$$

$$\boxed{\frac{1}{n}(X^T X)\alpha = \lambda\alpha}$$

Any benefits?

Nonlinear PCA using Kernels

Next consider the feature space: $\phi : x \to \phi(x)$

$$S^{\phi} = \frac{1}{n} X^{\phi} \left(X^{\phi}\right)^{T}, \text{ where } X^{\phi} = [\mathrm{x}_1^{\phi}, \mathrm{x}_2^{\phi}, \cdots, \mathrm{x}_\mathrm{n}^{\phi}].$$

$$v = \sum_i \alpha_i \phi(x_i) = X^{\phi} \alpha$$

$$\frac{1}{n} \left(X^{\phi}\right)^{T} X^{\phi} \alpha = \lambda \alpha$$

The (i,j)-th entry of $\left(X^{\phi}\right)^{T} X^{\phi}$ is $\phi(x_i) \bullet \phi(x_j)$

Apply the kernel trick: $K(x_i, x_j) = \phi(x_i) \bullet \phi(x_j)$

K is called the kernel matrix.

$$\boxed{\frac{1}{n} K \alpha = \lambda \alpha}$$

Nonlinear PCA using Kernels

- Projection of a test point x onto v:

$$\phi(x) \bullet v = \phi(x) \bullet \sum_i \alpha_i \phi(x_i)$$

$$= \sum_i \alpha_i \phi(x) \bullet \phi(x_i) = \sum_i \alpha_i K(x, x_i)$$

Explicit mapping is not required here.

Reference

- *Principal Component Analysis***.** I.T. Jolliffe.
- *Kernel Principal Component Analysis*. Schölkopf, et al.
- *Geometric Methods for Feature Extraction and Dimensional Reduction. Burges.*