Machine Learning

Deniz Gündüz and Krystian Mikolajczyk

Department of Electrical and Electronic Engineering
Imperial College London

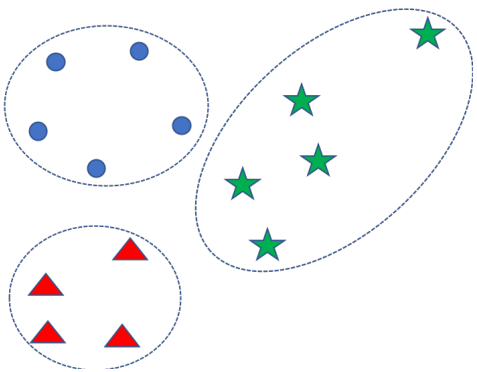
- Unsupervised learning
- Linear autoencoder
- Principal component analysis (PCA)

Machine Learning

- Supervised learning: Given data samples with labels $(\boldsymbol{x}, y)$, we want to learn a function $y = f(\boldsymbol{x})$ to predict labels of new samples
  - Predictive analytics
  - Classification: y is discrete
  - Regression: y is continuous
- Unsupervised learning: We are given only samples of data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, we want to compute a function $y = f(\boldsymbol{x})$ that provides a simpler representation
  - Descriptive analytics
  - y is discrete: clustering
  - y is continuous: dimension reduction, autoencoders

Unsupervised Learning

- Data without labels: $x_1, \ldots, x_n$
- Dimension reduction
  - Preprocessing step for supervised learning
  - Visualisation



- Clustering
  - Data compression
  - Group data samples that share "similarity"

Spanning Set

- Data points: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d$
- Assume *mean-centered* data points: We can subtract the mean in each dimension if they are not - reversible operation
- Consider set of basis vectors $\mathcal{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k\}$, $\boldsymbol{c}_j \in \mathbb{R}^d$, $\|\boldsymbol{c}_j\|^2 = 1$, $\forall j$
- $\mathcal{C}$ is called a spanning set if each point $\boldsymbol{x}_i$ can be written as some linear combination of basis vectors in $\mathcal{C}$, i.e.,

$$\boldsymbol{x}_i = \sum_{j=1}^{k} w_{i,j} \boldsymbol{c}_j$$

  for some $w_{i,1}, \ldots, w_{i,K}$.
- We can write more compactly as $C\boldsymbol{w}_i = \boldsymbol{x}_i$, where

$$C = [\boldsymbol{c}_1 \cdots \boldsymbol{c}_k], \qquad \text{and} \qquad \boldsymbol{w}_i = [w_{i,1}, \ldots, w_{i,k}]^T$$

- $\boldsymbol{w}_i$ is called encoding of $\boldsymbol{x}_i$ over C, while $C\boldsymbol{w}_i$ is called decoding

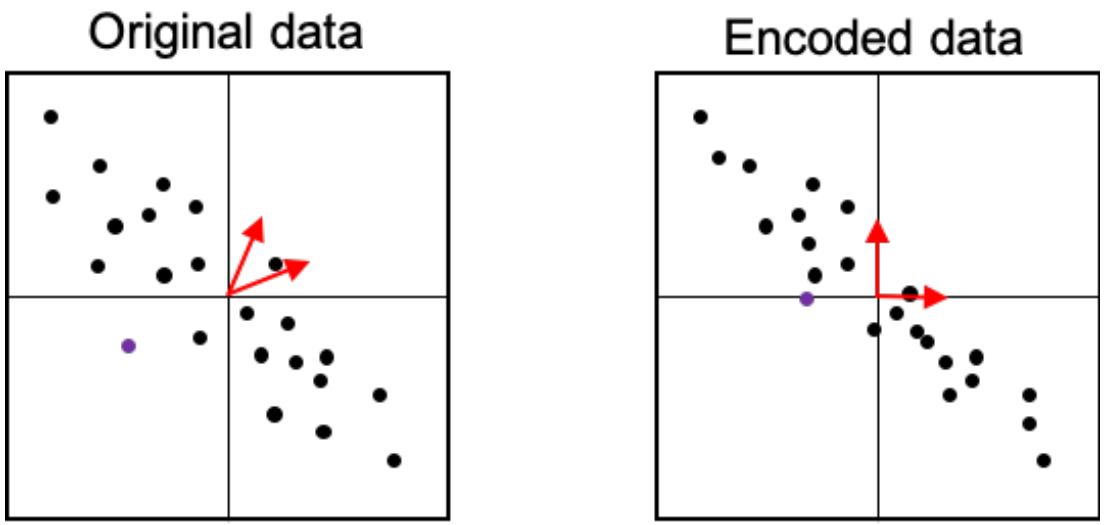
Spanning Set

If

1. $k = d$, and
2. vectors in $\mathcal{C}$ are linearly independent,

then $\mathcal{C}$ spans $\mathbb{R}^d$, i.e., any $\boldsymbol{x} \in \mathbb{R}^d$ can be written as $C\boldsymbol{w} = \boldsymbol{x}$ for some $\boldsymbol{w}$.

Hence, $\boldsymbol{x}_i \to \boldsymbol{w}_i$ is simply a coordinate rotation



Orthonormal Basis

- Set $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k\}$ is an orthonormal basis if

$$\boldsymbol{c}_i^T \boldsymbol{c}_j = \begin{cases} 1 & \text{if } i = j; \\ 0 & \text{if } i \neq j. \end{cases}$$

- Equivalently: $C^T C = I_{d \times d}$

- Encoding is simply: $\boldsymbol{w}_i = C^T \boldsymbol{x}_i$
- Autoencoder formula:

$$C \underbrace{C^T \boldsymbol{x}_i}_{\boldsymbol{w}_i} = \boldsymbol{x}_i$$

Imperfect Representation of Data

- If $k < d$, we cannot represent every possible data point in $\mathbb{R}^d$. We want $C\boldsymbol{w}_i \approx \boldsymbol{x}_i$
- We try to minimize the average error after decoding, i.e.,

$$\|C\boldsymbol{w}_i - \boldsymbol{x}_i\|^2$$

- If C is orthonormal, taking the derivative w.r.t. $\boldsymbol{w}_i$, we find

$$w_{i,j} = \boldsymbol{c}_j^T \boldsymbol{x}_i$$

Learning a Spanning Set

- Typically we are not given a spanning set, but need to learn from data
- Given dataset $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, we want to find the spanning set $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k\}$ that minimizes the average error

$$\min_{C, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_n} \frac{1}{n} \sum_{i=1}^{n} \|C\boldsymbol{w}_i - \boldsymbol{x}_i\|^2$$

- Not a convex problem, but can be solved using gradient descent

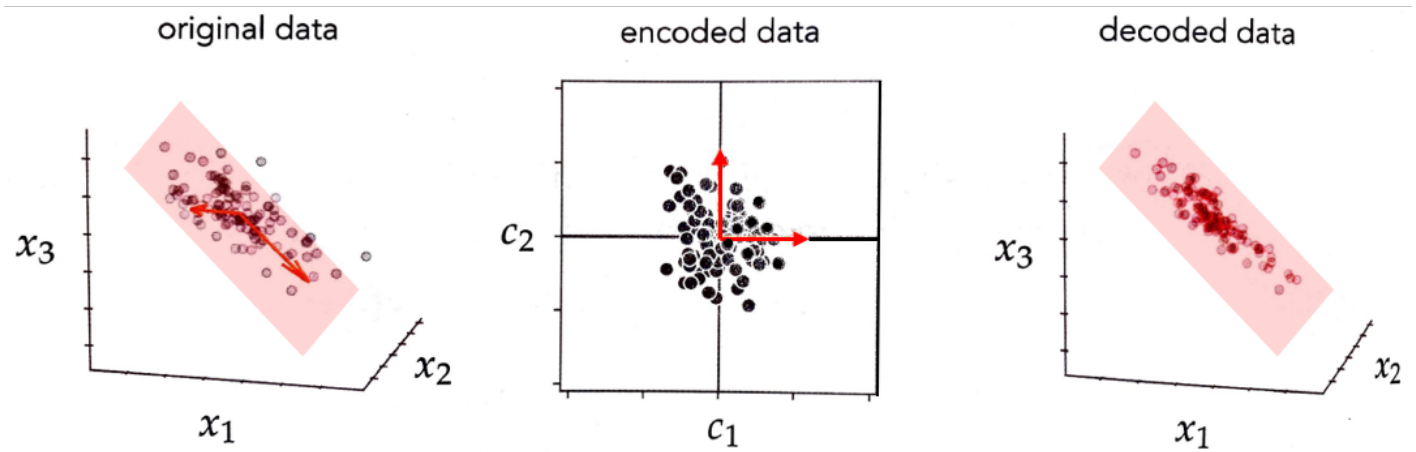


Figure from Watt et al., Machine Learning Refined.

Linear Autoencoder

- Remember: for orthonormal spanning vectors, we have $\boldsymbol{w}_i = C^T \boldsymbol{x}_i$
- Optimization problem can be written as

$$\min_{C : C^T C = I_{d \times d}} \frac{1}{n} \sum_{i=1}^{n} \| C C^T \boldsymbol{x}_i - \boldsymbol{x}_i \|^2$$

- No need to impose orthonormality

Optimal Linear Autoencoder

$$\min_{C} \frac{1}{n} \sum_{i=1}^{n} \| CC^T \boldsymbol{x}_i - \boldsymbol{x}_i \|^2$$

- CC^T is a symmetric $d \times d$ matrix
- Eigenvalue decomposition: $CC^T = VDV^T$, where $V \in \mathbb{R}^{d \times d}$ is an orthonormal matrix of eigenvectors (i.e., $VV^T = I_{d \times d}$), and $D \in \mathbb{R}^{d \times d}$ is a diagonal matrix with at most k nonnegative eigenvalues

Optimal Linear Autoencoder

$$\begin{aligned}\|CC^T\boldsymbol{x}_i - \boldsymbol{x}_i\|^2 &= \|VDV^T\boldsymbol{x}_i - \boldsymbol{x}_i\|^2 \\ &= \boldsymbol{x}_i^T VDDV^T\boldsymbol{x}_i - 2\boldsymbol{x}_i^T VDV^T\boldsymbol{x}_i + \boldsymbol{x}_i^T\boldsymbol{x}_i \\ &= \boldsymbol{x}_i^T VDDV^T\boldsymbol{x}_i - 2\boldsymbol{x}_i^T VDV^T\boldsymbol{x}_i + \boldsymbol{x}_i^T VV^T\boldsymbol{x}_i\end{aligned}$$

Define $\boldsymbol{q}_i = V^T\boldsymbol{x}_i$. Above can be rewritten as

$$\begin{aligned}\boldsymbol{q}_i^T DD\boldsymbol{q}_i - 2\boldsymbol{q}_i^T D\boldsymbol{q}_i + \boldsymbol{q}_i^T\boldsymbol{q}_i &= \boldsymbol{q}_i^T(D^2 - 2D + I_{d\times d})\boldsymbol{q}_i \\ &= \boldsymbol{q}_i^T(D - I_{d\times d})^2\boldsymbol{q}_i\end{aligned}$$

Optimization problem

$$\min_C \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{q}_i^T(D - I_{d\times d})^2\boldsymbol{q}_i$$

This is minimized when D has k eigenvalues of $+1$. This means that C is an orthonormal matrix.

Principal Component Analysis (PCA)

- Widely used for dimensionality reduction, lossy data compression, feature extraction, and data visualization
- Two formulations possible, leading to the same solution:
  - Maximum variance formulation: Orthogonal projection of data to a lower dimensional principal subspace, such that the variance of projected data is maximized (Hotelling'33)
  - Minimum error formulation: Linear projection that minimizes the mean squared error between data points and their projections (Pearson'01)

PCA - Maximum Variance Formulation

Consider $k = 1$. We want a single vector $\boldsymbol{c}$ to project on with $\|\boldsymbol{c}\|^2 = 1$.

We want to maximize

$$\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{c}^T\boldsymbol{x}_i - \boldsymbol{c}^T\bar{\boldsymbol{x}}\right)^2$$

where $\bar{\boldsymbol{x}}$ is the mean-value of data, which we assumed to be zero. Variance reduces to

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{c}^T\boldsymbol{x}_i\right)^2 &= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{c}^T(\boldsymbol{x}_i\boldsymbol{x}_i^T)\boldsymbol{c} \\ &= \boldsymbol{c}^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T\right)\boldsymbol{c} \\ &= \boldsymbol{c}^T\left(\frac{1}{n}XX^T\right)\boldsymbol{c}\end{aligned}$$

where $X = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_n]$ is the data matrix.

PCA - Minimum Error Formulation

Assume orthonormal spanning set with k elements. We have

$$\begin{aligned}\|CC^T\boldsymbol{x}_i - \boldsymbol{x}_i\|^2 &= \boldsymbol{x}_i^T CC^T CC^T \boldsymbol{x}_i - 2\boldsymbol{x}_i^T CC^T \boldsymbol{x}_i + \boldsymbol{x}_i^T \boldsymbol{x}_i \\ &= -\boldsymbol{x}_i^T CC^T \boldsymbol{x}_i + \boldsymbol{x}_i^T \boldsymbol{x}_i \\ &= -\|C^T \boldsymbol{x}_i\|^2 + \|\boldsymbol{x}_i\|^2\end{aligned}$$

Since $\boldsymbol{x}_i$ is given, we only minimize $-\|C^T \boldsymbol{x}_i\|^2 = -\sum_{j=1}^{k} \left(\boldsymbol{c}_j^T \boldsymbol{x}_i\right)^2$

Hence, the objective function is

$$\min_{\substack{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_k: \\ \|\boldsymbol{c}_i\|^2=1, \\ \boldsymbol{c}_i^T \boldsymbol{c}_j=0, i\neq j}} -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}\left(\boldsymbol{c}_j^T \boldsymbol{x}_i\right)^2$$

We can minimize one basis vector at a time!

Principal Component Analysis (PCA)

We begin with $\boldsymbol{c}_1$. We want

$$\min_{\boldsymbol{c}_1 : \|\boldsymbol{c}_1\|^2 = 1} -\frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{c}_1^T \boldsymbol{x}_i \right)^2$$

Note that, this measure the variance of data along the direction of $\boldsymbol{c}_1$ (because we assumed zero-mean data): Equivalent to max. variance formulation

We have

$$\begin{aligned}
\frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{c}_1^T \boldsymbol{x}_i \right)^2 &= \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{c}_1^T \boldsymbol{x}_i \boldsymbol{x}_i^T \boldsymbol{c}_1 \\
&= \boldsymbol{c}_1^T \left(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T \right) \boldsymbol{c}_1 \\
&= \boldsymbol{c}_1^T \left(\frac{1}{n} X X^T \right) \boldsymbol{c}_1
\end{aligned}$$

Principal Component Analysis (PCA)

$$\begin{aligned} \min_{\boldsymbol{c}_1} \quad & -\boldsymbol{c}_1^T \left(\frac{1}{n} X X^T\right) \boldsymbol{c}_1 \\ \text{s.t.} \quad & \boldsymbol{c}_1^T \boldsymbol{c}_1 = 1 \end{aligned}$$

We form the Lagrangian

$$\mathcal{L}(\boldsymbol{c}_1, \alpha) = -\boldsymbol{c}_1^T \left(\frac{1}{n} X X^T\right) \boldsymbol{c}_1 + \alpha(\boldsymbol{c}_1^T \boldsymbol{c}_1 - 1)$$

We have $$\frac{\partial \mathcal{L}}{\partial \boldsymbol{c}_1} = -2 \left(\frac{1}{n} X X^T\right) \boldsymbol{c}_1 + 2\alpha \boldsymbol{c}_1 = 0$$

We get $\left(\frac{1}{n} X X^T\right) \boldsymbol{c}_1 = \alpha \boldsymbol{c}_1$

Implies that α is an eigenvalue of $\left(\frac{1}{n} X X^T\right)$ with associated eigenvector $\boldsymbol{c}_1$.

Principal Component Analysis (PCA)

$$\begin{aligned} \min_{\boldsymbol{c}_1} \quad & -\boldsymbol{c}_1^T \left(\frac{1}{n} X X^T\right) \boldsymbol{c}_1 \\ \text{s.t.} \quad & \boldsymbol{c}_1^T \boldsymbol{c}_1 = 1 \end{aligned}$$

The optimal value is obtained as

$$\boldsymbol{c}_1^T \left(\frac{1}{n} X X^T\right) \boldsymbol{c}_1 = \boldsymbol{c}_1^T \alpha \boldsymbol{c}_1 = \alpha$$

We should choose $\boldsymbol{c}_1$ as the eigenvector corresponding to the largest eigenvalue of $\left(\frac{1}{n} X X^T\right)$. This is called the first principal component.

Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$ be the eigenvalues of $\frac{1}{n} X X^T$ with corresponding eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$.

Approximation with First Principal Component

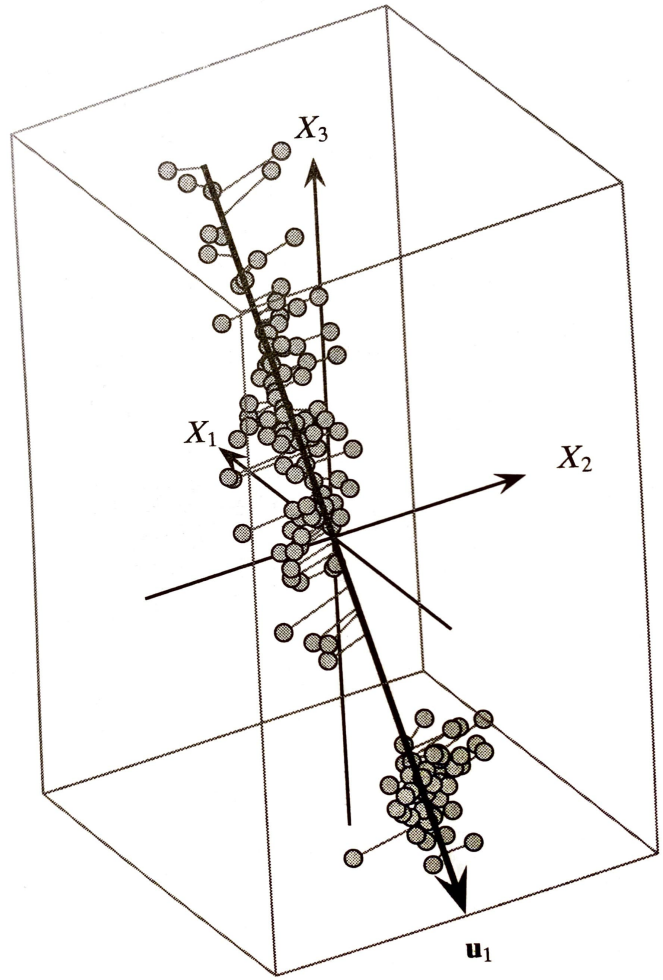


Figure from Zakir and Meira, Data Mining and Machine Learning.

Second Principal Component

$$\begin{aligned} \min_{\boldsymbol{c}_2} \quad & -\boldsymbol{c}_2^T \left(\frac{1}{n} X X^T\right) \boldsymbol{c}_2 \\ \text{s.t.} \quad & \boldsymbol{c}_2^T \boldsymbol{c}_2 = 1 \text{ and } \quad \boldsymbol{c}_2^T \boldsymbol{u}_1 = 0 \end{aligned}$$

$$\mathcal{L}(\boldsymbol{c}_2, \theta, \beta) = -\boldsymbol{c}_2^T \left(\frac{1}{n} X X^T\right) \boldsymbol{c}_2 + \theta(\boldsymbol{c}_2^T \boldsymbol{c}_2 - 1) + \beta \boldsymbol{c}_2^T \boldsymbol{u}_1$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{c}_2} = -2 \left(\frac{1}{n} X X^T\right) \boldsymbol{c}_2 + 2\theta \boldsymbol{c}_2 + \beta \boldsymbol{u}_1 = 0$$

Multiplying both sides with $\boldsymbol{u}_1^T$, we get

$$-2\boldsymbol{u}_1^T \left(\frac{1}{n} X X^T\right) \boldsymbol{c}_2 + 2\theta \boldsymbol{u}_1^T \boldsymbol{c}_2 + \beta \boldsymbol{u}_1^T \boldsymbol{u}_1 = 0$$

$$\beta = 2\boldsymbol{c}_2^T \left(\frac{1}{n} X X^T\right) \boldsymbol{u}_1 = 2\boldsymbol{c}_2^T \lambda_1 \boldsymbol{u}_1 = 0$$

Second Principal Component

Plugging $\beta = 0$ into $\frac{\partial \mathcal{L}}{\partial \boldsymbol{c}_2} = 0$ above, we get

$$\left(\frac{1}{n} X X^T\right) \boldsymbol{c}_2 = \theta \boldsymbol{c}_2$$

Implies that $\boldsymbol{c}_2$ is another eigenvector of $\frac{1}{n} X X^T$, orthogonal to $\boldsymbol{u}_1$.

The maximum value is given by θ, its eigenvalue. Hence, the second principal component is given by $\boldsymbol{c}_2 = \boldsymbol{u}_2$.

We can continue similarly, and the jth principal component is given by the eigenvector corresponsing to the jth largest eigenvalue of the mean-centered data covariance matrix $\frac{1}{n} X X^T$.

Approximation with Two Principal Components

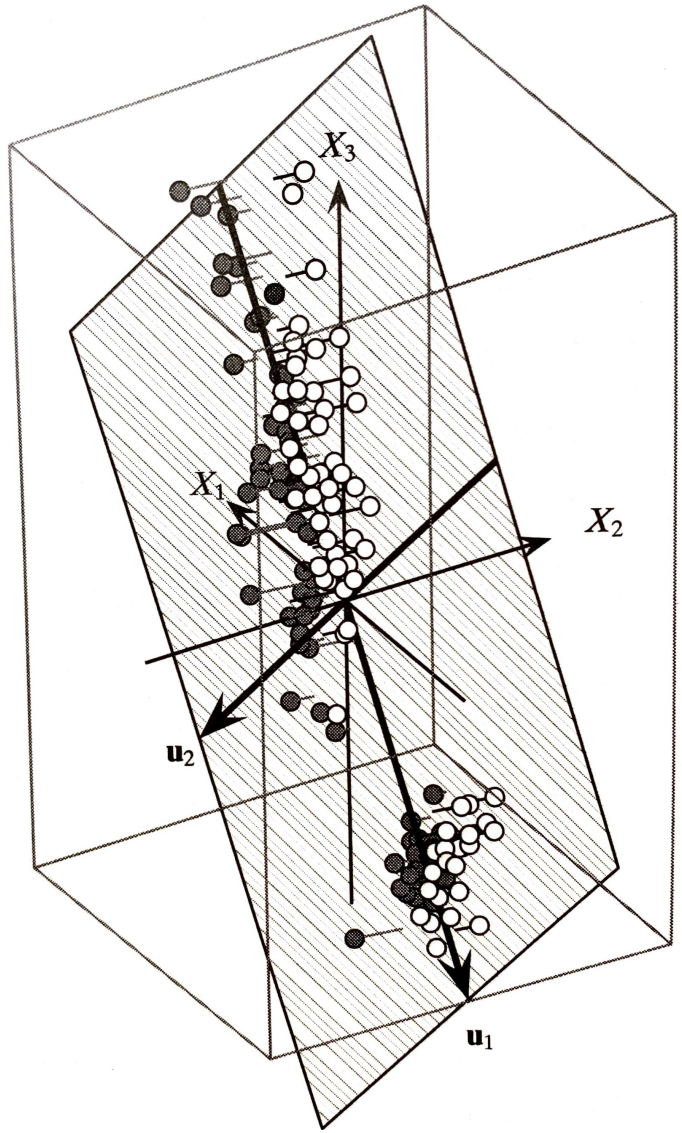


Figure from Zakir and Meira, Data Mining and Machine Learning.

Application of PCA

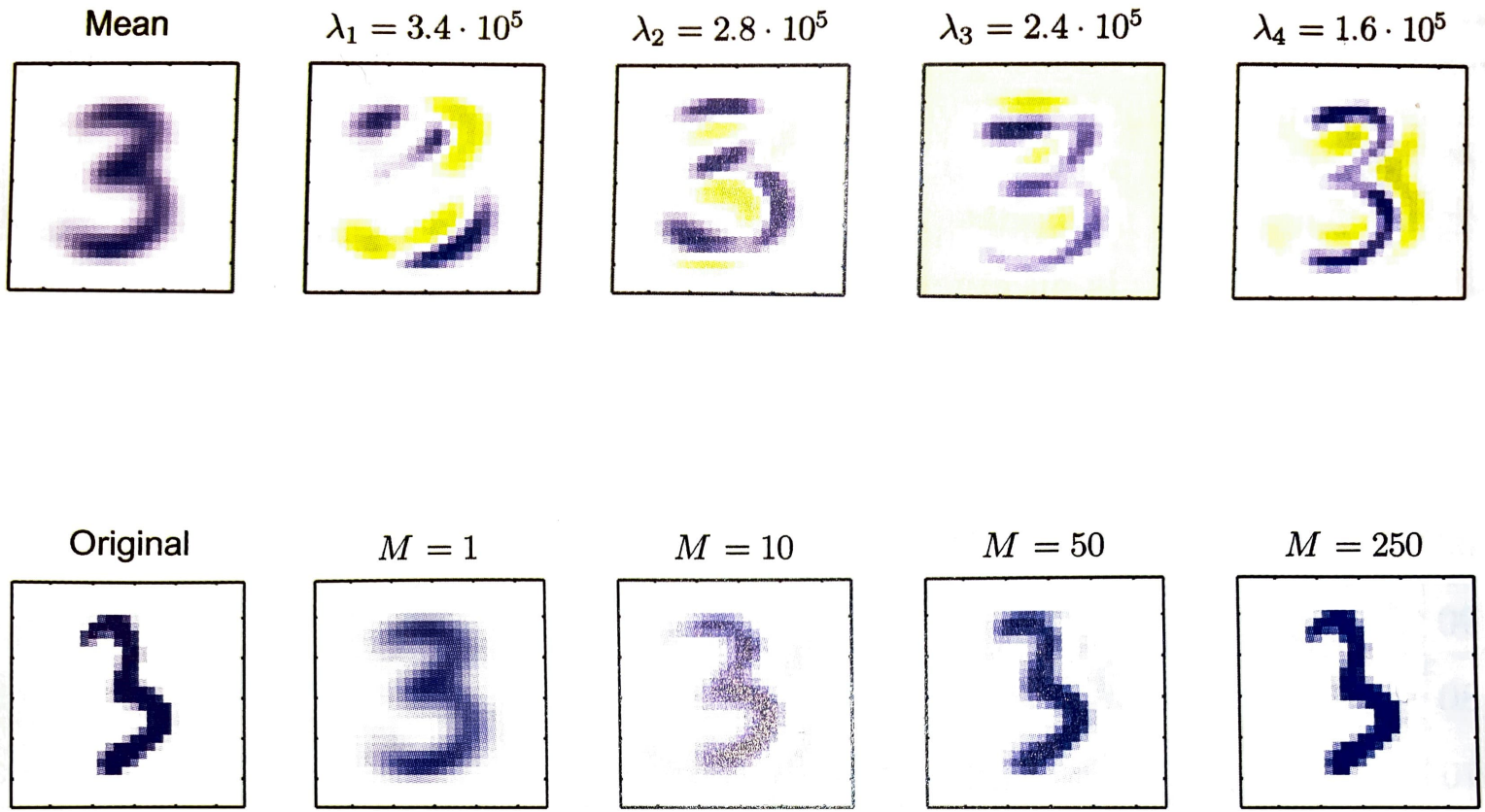


Figure from Bishop, Pattern Recognition and Machine Learning.

Application of PCA - Whitening

- Pre-process data to standardize some properties
- Let $S = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T$ be the data covariance matrix ($\bar{\boldsymbol{x}}$ is the sample mean) with eigenvalue decomposition

$$S = ULU^T$$

  where U is the orthogonal matrix with columns $\boldsymbol{u}_i$, and L is the diagonal matrix with elements λ_i.
- Transform each data points as

$$\boldsymbol{y}_i = L^{-1/2}U^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}})$$

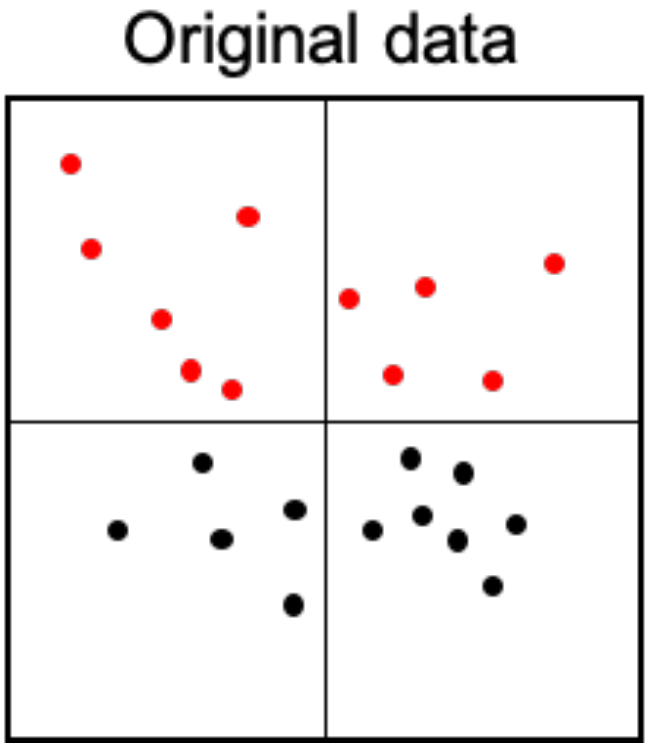
Application of PCA - Whitening

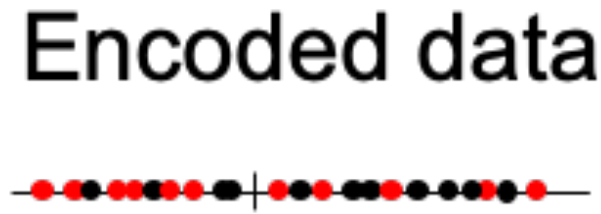
We have

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_i^T &= \frac{1}{n}\sum_{i=1}^{n} L^{-1/2} U^T (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T U L^{-1/2} \\
&= L^{-1/2} U^T \left(\frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \right) U L^{-1/2} \\
&= L^{-1/2} U^T S U L^{-1/2} \\
&= L^{-1/2} U^T U L U^T U L^{-1/2} \\
&= L^{-1/2} L L^{-1/2} = I_{d \times d}
\end{aligned}$$

Each component of the new data representation are perfectly uncorrelated!

Classification with PCA





[Additional Slides]

Variance in PCA

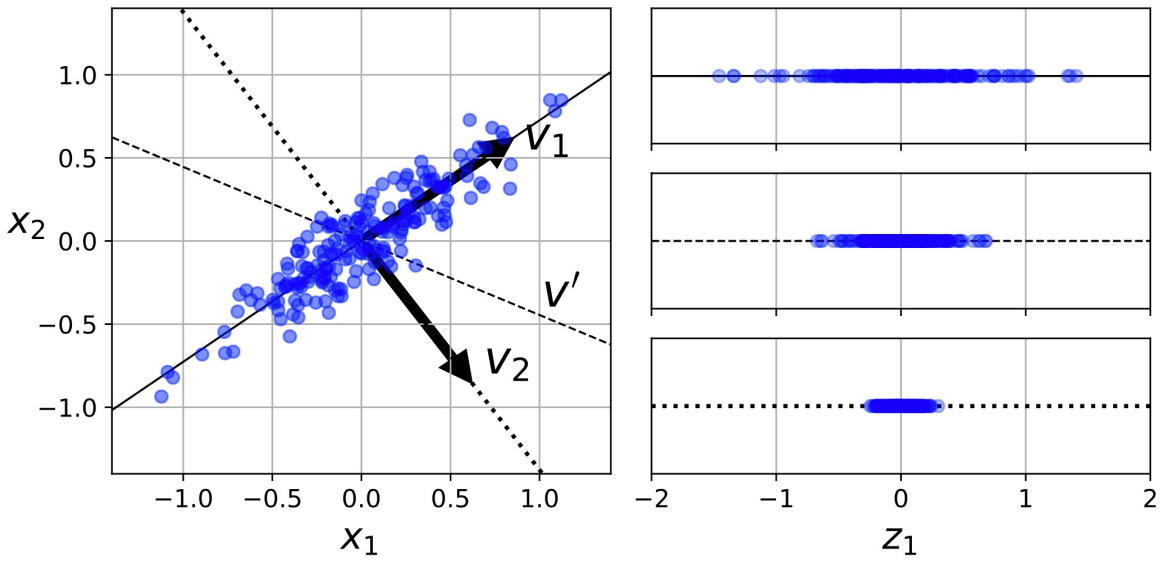
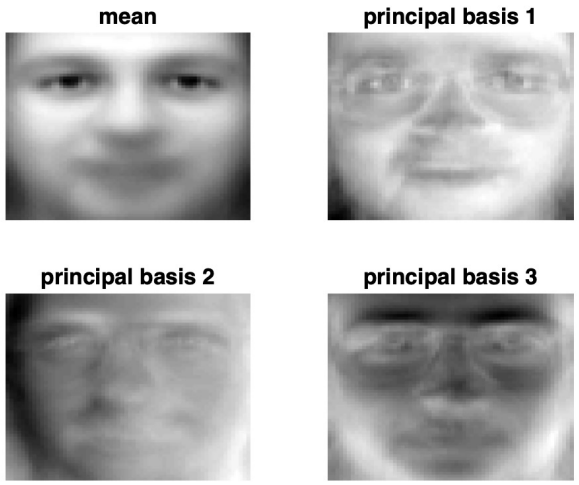


Figure 20.4: Illustration of the variance of the points projected onto different 1d vectors. v_1 is the first principal component, which maximizes the variance of the projection. v_2 is the second principal component which is direction orthogonal to v_1. Finally v' is some other vector in between v_1 and v_2. Adapted from Figure 8.7 of [Gér19]. Generated by code at figures.probml.ai/book1/20.4

[Additional Slides]

Example

(a)



(b)

Figure 20.3: a) Some randomly chosen 64×64 pixel images from the Olivetti face database. (b) The mean and the first three PCA components represented as images. Generated by code at figures.probml.ai/book1/20.3.

[Additional Slides]

Why do we scale/normalize/standardize our data?

1. Scaling does not affect our statistical inference.

2. When we make new features out of the existing ones (e.g., by adding or multiplying two feature), then if features are not in the same scale, one of them can cancel out the other one.

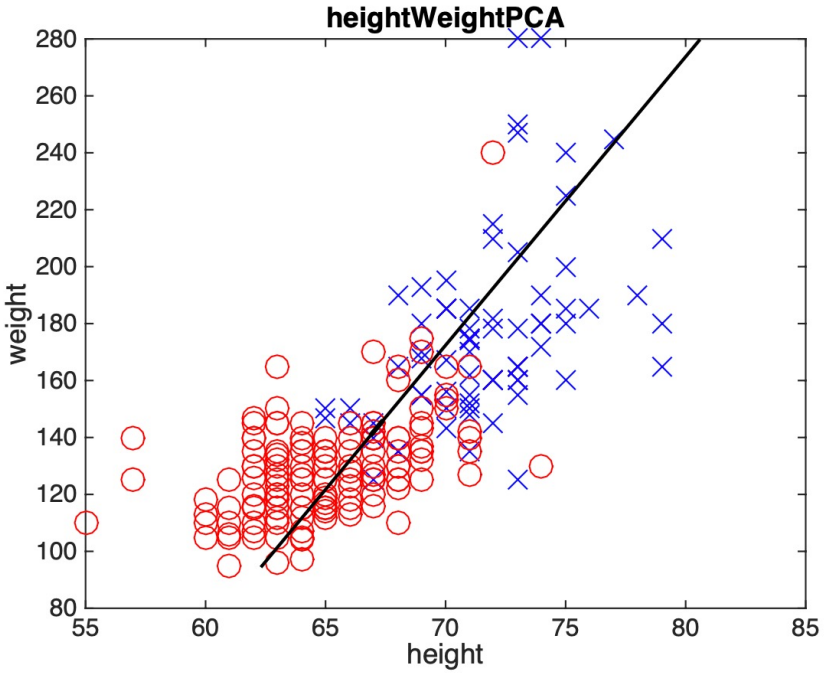
3. PCA can only be interpreted as the singular value decomposition of a data matrix when the columns have first been centered by their means.

4. Simplification of analysis:
the sample covariance matrix of a matrix of values centered by their sample means is simply $X'X$. Similarly, if a univariate random variable X has been mean centered, then var(X)=$E(X2)$

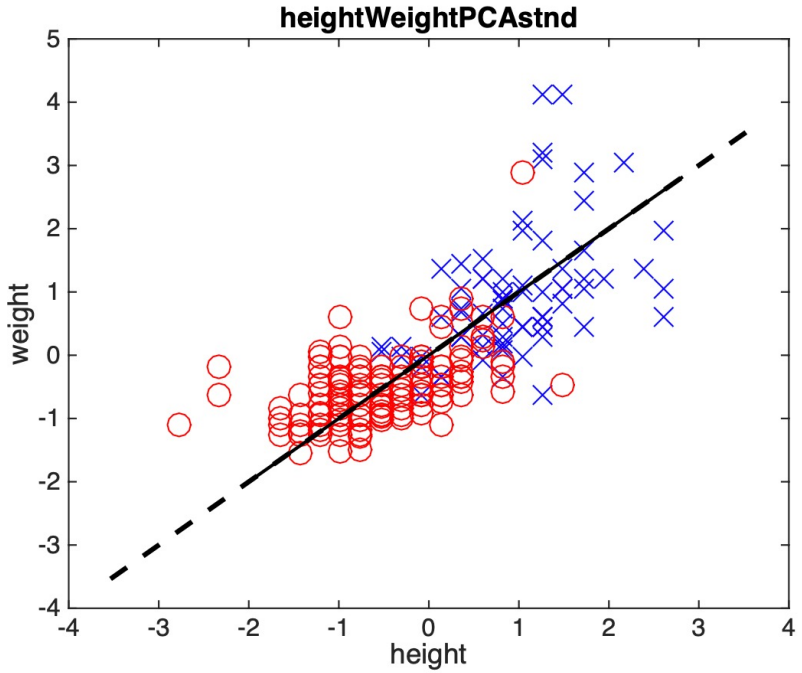
https://stats.stackexchange.com/questions/29781/when-conducting-multiple-regression-when-should-you-center-your-predictor-varia

[Additional Slides]

Example



(a)



(b)

Figure 20.5: Effect of standardization on PCA applied to the height/weight dataset. (Red=female, blue=male.) Left: PCA of raw data. Right: PCA of standardized data. Generated by code at figures.probml.ai/book1/20.5.