EMS702P Statistical Thinking and Applied Machine Learning

Week 3.1 – Dimensionality Reduction & Principal Component Analysis

Jun Chen

Introduction to Statistics and Machine Learning

Edition: v1.1

Table of Contents

1 Dimensionality Reduction **1**

1.1 The Curse of Dimensionality 1

1.2 Dimensionality Reduction: Feature Extraction 2

1.3 Vector Representation 3

2 Principal Component Analysis (PCA) **5**

2.1 PCA Steps 5

2.2 Interpretation of PCA 7

2.3 An Example 8

3 PCA Practices **10**

3.1 Selection Criteria 10

3.2 Data Normalisation 10

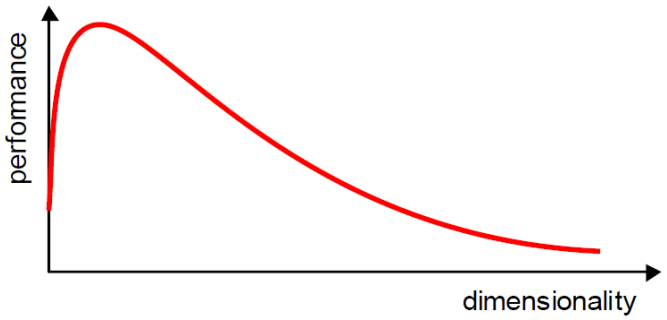
3.3 Advantages vs. Disadvantages 11

4 Further Reading **12**

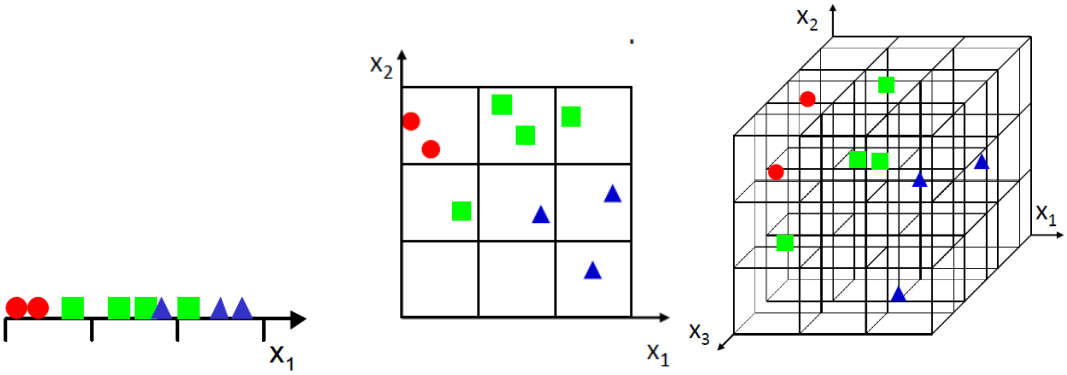
1 Dimensionality Reduction

1.1 The Curse of Dimensionality

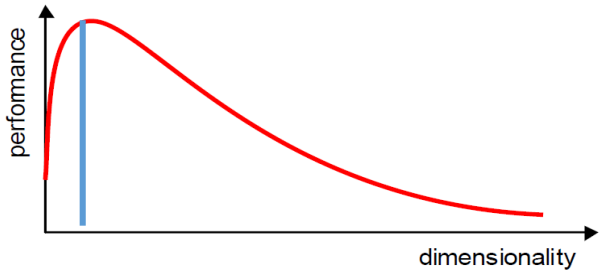
- A large number of features will not always improve classification accuracy.
- It may actually lead to worse performance.



- The number of training examples required increases exponentially with dimensionality.



- The objective of dimensionality reduction is to choose an optimum set of features of lower dimensionality to improve the model performance.



- Different methods can be used to reduce dimensionality: Feature extraction and Feature selection.
- Feature extraction: constructs new features (i.e., through some function $f()$) from the existing features.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix} \xrightarrow{f(\mathbf{x})} \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_K \end{bmatrix}$$

- Feature selection: chooses a subset of the original features (e.g. t-test).

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix} \rightarrow \mathbf{y} = \begin{bmatrix} x_{i_1} \\ x_{i_2} \\ . \\ . \\ x_{i_K} \end{bmatrix}$$

1.2 Dimensionality Reduction: Feature Extraction

Linear combinations are particularly attractive because they are simpler to compute and analytically tractable.

Given $X \in R^N$, find a $K \times N$ matrix T, where $K \ll N$, such that

$$Y = TX$$

This is a projection from the N-dimensional space to an K-dimensional space. From a mathematical point of view, finding an optimum mapping $\boldsymbol{y} = f(\boldsymbol{x})$ is equivalent to optimizing an **objective** criterion.

Different methods use different objective criteria, e.g.,

- Minimise Information Loss: represent the data as accurately as possible in the lower-dimensional space.
- Maximize Discriminatory Information: enhance the class-discriminatory information in the lower-dimensional space.

Popular linear feature extraction methods:

- **Principal Components Analysis (PCA):** Seeks a projection that preserves as much information in the data as possible.
- **Linear Discriminant Analysis (LDA):** Seeks a projection that best discriminates the data.

Many other methods:

- Making features as independent as possible (Independent Component Analysis or ICA).
- Retaining interesting directions (Projection Pursuit).
- Embedding to lower dimensional manifolds (Isomap, Locally Linear Embedding or LLE).

1.3 Vector Representation

For a vector $\mathbf{x} \in R^n$ can be represented by n components:

$$\mathbf{x}: \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix}$$

Assuming the standard base <v_1, v_2, …, v_N> (i.e., unit vectors in each dimension), x_i can be obtained by projecting **x** along the direction of v_i:

$$x_i = \frac{\mathbf{x}^T v_i}{v_i^T v_i} = \mathbf{x}^T v_i$$

x can be "reconstructed" from its projections as follows:

$$\mathbf{x} = \sum_{i=1}^{N} x_i v_i = x_1 v_1 + x_2 v_2 + \ldots + x_N v_N$$

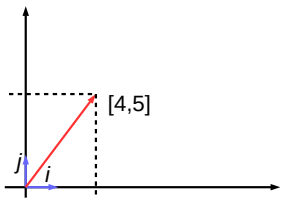
Since the basis vectors are the same for all x ϵ R^n (standard basis), we typically represent them as a n-component vector.

Example 1

Assuming n=2, **x** is a vector as below.

$$\mathbf{x}: \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

Assuming the standard base <v_1=i, v_2=j>, x_i can be obtained by projecting x along the direction of v_i:



$$x_1 = \mathbf{x}^T i = \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 3 \quad x_2 = \mathbf{x}^T j = \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 4$$

x can be "reconstructed" from its projections as follows:

$$\mathbf{x} = 3i + 4j$$

2 Principal Component Analysis (PCA)

If $\mathbf{x} \in R^N$, then it can be written as a linear combination of an *orthonormal* set of N basis vectors $<v_1, v_2, \ldots, v_N>$ in R^N (e.g., using the *standard base*):

$$v_i^T v_j = \begin{cases} 1 & if \ \ i = j \\ 0 & otherwise \end{cases} \qquad \mathbf{x} = \sum_{i=1}^{N} x_i v_i = x_1 v_1 + x_2 v_2 + \ldots + x_N v_N$$

$$where \quad x_i = \frac{\mathbf{x}^T v_i}{v_i^T v_i} = \mathbf{x}^T v_i \qquad \mathbf{x}: \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix}$$

PCA seeks to approximate **x** in a subspace of R^N using a new set of K<<N basis vectors $<u_1, u_2, \ldots, u_K>$ in R^N:

$$\hat{\mathbf{x}} = \sum_{i=1}^{K} y_i u_i = y_1 u_1 + y_2 u_2 + \ldots + y_K u_K$$

(reconstruction)

$$where \quad y_i = \frac{\mathbf{x}^T u_i}{u_i^T u_i} = \mathbf{x}^T u_i \qquad \hat{\mathbf{x}}: \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_K \end{bmatrix}$$

such that $||\mathbf{x} - \hat{\mathbf{x}}||$ is minimised, i.e., minimise the information loss.

2.1 PCA Steps

The "optimal" set of basis vectors $<u_1, u_2, \ldots, u_K>$ can be found as follows:

(1) Find the eigenvectors u_i of the covariance matrix of the (training) data $\Sigma_{\mathbf{x}}$

$$\Sigma_{\mathbf{x}} u_i = \lambda_i u_i$$

(2) Choose the K "largest" eigenvectors u_i (i.e., corresponding to the K "largest" eigenvalues λ_i), where $<u_1, u_2, \ldots, u_K>$ corresponds to the "optimal" basis.

We refer to the **"largest" eigenvectors** u_i as **principal components.**

Suppose we are given $x_1, x_2, \ldots, x_M$ (N x 1) vectors, where N is the number of features, and M is the number of data points in a sample.

Step 1: compute the sample mean

$$\bar{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}_i$$

Step 2: subtract sample mean (i.e., center data at zero)

$$\Phi_i = \mathbf{x}_i - \bar{\mathbf{x}}$$

Step 3: compute the sample covariance matrix Σ_x

$$\Sigma_{\mathbf{x}} = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T = \frac{1}{M}\sum_{i=1}^{M}\Phi_i\Phi_i^T$$

Step 4: compute the eigenvalues/eigenvectors of Σ_x

$$\Sigma_x u_i = \lambda_i u_i$$

where we assume

$$\lambda_1 > \lambda_2 > \ldots > \lambda_N$$

Since Σ_x is symmetric, $<u_1, u_2, \ldots, u_N>$ form an orthogonal basis in R^N and we can represent any $\mathbf{x} \in R^N$ as:

$$\mathbf{x} - \bar{\mathbf{x}} = \sum_{i=1}^{N} y_i u_i = y_1 u_1 + y_2 u_2 + \ldots + y_N u_N$$

$$y_i = \frac{(\mathbf{x} - \bar{\mathbf{x}})^T u_i}{u_i^T u_i} = (\mathbf{x} - \bar{\mathbf{x}})^T u_i \qquad if \;\; \| u_i \| = 1$$

$$\mathbf{x}-\bar{\mathbf{x}}:\begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ . \\ . \\ y_N \end{bmatrix}$$

Step 5: dimensionality reduction step – approximate **x** using only the first K eigenvectors (K<<N) (i.e., corresponding to the K largest eigenvalues where K is a parameter).

$$\hat{\mathbf{x}}-\bar{\mathbf{x}}=\sum_{i=1}^{K} y_i u_i = y_1 u_1 + y_2 u_2 + \ldots + y_K u_K$$

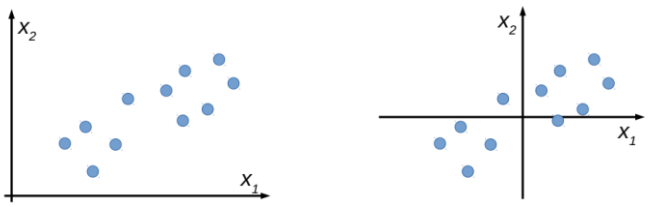
$$\mathbf{x}-\bar{\mathbf{x}}:\begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ . \\ . \\ x_N \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ . \\ . \\ y_N \end{bmatrix} \rightarrow \hat{\mathbf{x}}-\bar{\mathbf{x}}:\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_K \end{bmatrix}$$

Note that if K=N, then i.e., zero reconstruction error.

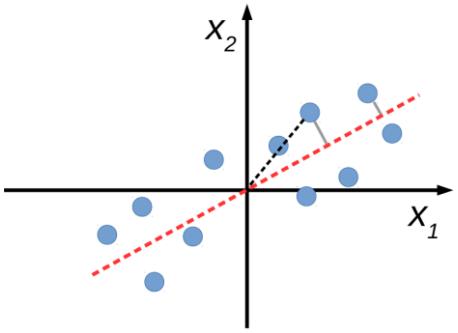
PCA is the *linear transformation* **y = Tx,** where $T=U^T=\langle u_1, u_2, \ldots, u_K\rangle^T$ and is an KxN matrix.

2.2 Interpretation of PCA

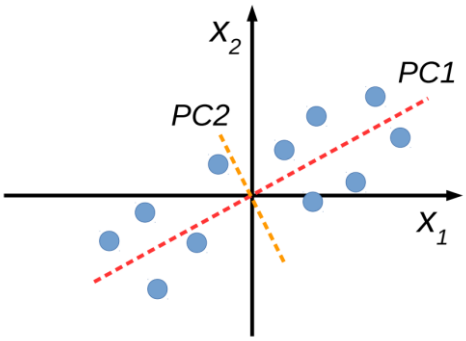
Step 1: Subtracting the sample mean centres the data



Step 2: Principal component is the line of best fit, passing through the origin, i.e. minimise the sum of the distances from the points to the line.



Step 3: Finding other principal components, each of which always explains some proportion of the total variance in the data.



2.3 An Example

Compute the PCA of the following dataset:

(1,2), (3,3), (3,5), (5,4), (5,6), (6,5), (8,7), (9,8)

Step 1 find the mean.

Step 2 compute the covariance matrix.

x1	x2	x1-avg	x2-avg	(x1-avg)^2	(x1-avg)(x2-avg)	(x2-avg)^2
1	2	-4	-3	16	12	9
3	3	-2	-2	4	4	4
3	5	-2	0	4	0	0
5	4	0	-1	0	0	1
5	6	0	1	0	0	1
6	5	1	0	1	0	0
8	7	3	2	9	6	4
9	8	4	3	16	12	9
avg	avg			sum/n	sum/n	sum/n
5	5			6.25	4.25	3.5

$$\hat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t \quad \Sigma_x = \begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}$$

Step 3 compute the eigenvalues by finding the roots of the characteristic polynomial.

$$\Sigma_x v = \lambda v \Rightarrow |\Sigma_x - \lambda I| = 0$$

$$\begin{vmatrix} 6.25 - \lambda & 4.25 \\ 4.25 & 3.5 - \lambda \end{vmatrix} = 0$$

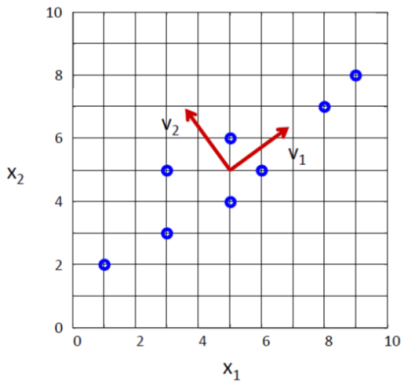
$$\lambda_1 = 9.34, \lambda_2 = 0.41$$

Step 4 the eigenvectors are the solutions of the systems

$$\Sigma_{\mathbf{x}} u_i = \lambda_i u_i$$

$$\begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}\begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} \lambda_1 v_{11} \\ \lambda_1 v_{12} \end{bmatrix} \Rightarrow \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} 0.81 \\ 0.59 \end{bmatrix}$$

$$\begin{bmatrix} 6.25 & 4.25 \\ 4.25 & 3.5 \end{bmatrix}\begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \begin{bmatrix} \lambda_2 v_{21} \\ \lambda_2 v_{22} \end{bmatrix} \Rightarrow \begin{bmatrix} v_{21} \\ v_{22} \end{bmatrix} = \begin{bmatrix} -0.59 \\ 0.81 \end{bmatrix}$$



Eigenvectors can be normalized to unit-length using:

$$\hat{v}_i = \frac{v_i}{\| v_i \|}$$

Questions: what is the meaning of eigenvectors?

The larger they are these absolute values, the more a specific feature contributes to that principal component.

3 PCA Practices

3.1 Selection Criteria

K is typically chosen based on how much information (variance) we want to preserve. we normally choose the smallest *k* that satisfies the following inequality:

$$\frac{\sum_{i=1}^{K}\lambda_i}{\sum_{i=1}^{N}\lambda_i} > T \quad \textit{where } T \textit{ is a threshold } (e.g., 0.9)$$

- If T=0.9, for example, we "preserve" 90% of the information (variance) in the data.
- If K=N, then we "preserve" 100% of the information in the data (i.e., just a "change" of basis and).

The approximation error (or reconstruction error) can be computed by:

$$|| \mathbf{x} - \hat{\mathbf{x}} ||$$

$$\hat{\mathbf{x}} = \sum_{i=1}^{K} y_i u_i + \overline{\mathbf{x}} = y_1 u_1 + y_2 u_2 + ... + y_K u_K + \overline{\mathbf{x}}$$

It can also be shown that the approximation error can be computed as follows:

$$|| \mathbf{x} - \hat{\mathbf{x}} || = \frac{1}{2} \sum_{i=K+1}^{N} \lambda_i$$

3.2 Data Normalisation

The principal components are dependent on the *units* used to measure the original variables as well as on the *range* of values they assume. Data should always be normalized prior to using PCA.

A common normalization method is to transform all the data to have zero mean and unit standard deviation:

$$x_i^{norm} = \frac{x_i - \mu_i}{\sigma_i}$$

where μ and σ are the sample mean and standard deviation of the *i*-th feature x_i.

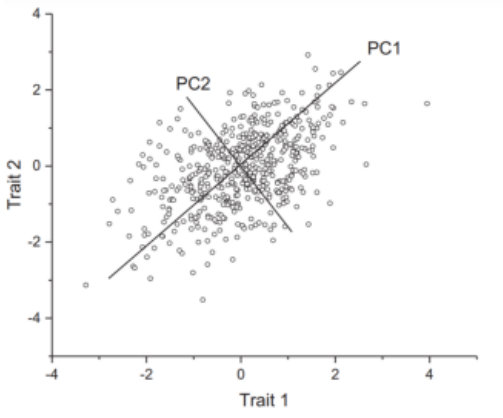
3.3 Advantages vs. Disadvantages

Advantages

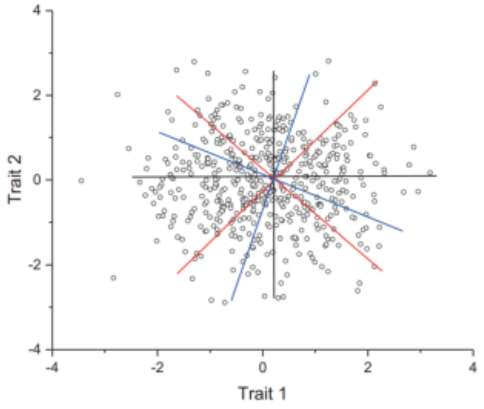
- Finds Correlated Features
- Improves Prediction Algorithm Performance
- Reduces Overfitting by decreasing dimensionality
- Improves Visualisation

Disadvantages

- PCA is meaningful only if there is linear correlation in features



Unique principal components.



No clear pattern of correlation since the data are spherical. Principal components are random axes.

- PCA is scale variant - needs normalisation
- less interpretability

- PCA has a hard time working with missing data and outliers

4 Further Readings

[1] StatQuest: Principal Component Analysis (PCA), Step-by-Step

https://www.youtube.com/watch?v=FgakZw6K1QQ