

**VIETNAM GENERAL CONFEDERATION OF LABOR
TON DUC THANG UNIVERSITY
INFORMATION TECHNOLOGY FACULTY**

MACHINE LEARNING
Assignment

DIMENSIONALITY REDUCTION

Instructor: **Prof. Lê Anh Cường**

Student 1: **Đỗ Phạm Quang Hưng - 520K0127**

Student 2: **Lê Phước Thịnh - 520K0343**

HO CHI MINH CITY, 2023

Table of Contents

Table of Contents	2
Dimensionality Reduction and PCA	3
Introduction	3
PCA - How to do	5
References	8

Dimensionality Reduction and PCA

Introduction

Dimensionality reduction refers to reducing the number of input variables for a dataset.

If your data is represented using rows and columns, such as in a spreadsheet, then the input variables are the columns that are fed as input to a model to predict the target variable. Input variables are also called features.

We can consider the columns of data representing dimensions on an n -dimensional feature space and the rows of data as points in that space. This is a useful geometric interpretation of a dataset.

“In a dataset with k numeric attributes, you can visualize the data as a cloud of points in k -dimensional space ...”

— Page 305, [Data Mining: Practical Machine Learning Tools and Techniques](#), 4th edition, 2016.

Having a large number of dimensions in the feature space can mean that the volume of that space is very large, and in turn, the points that we have in that space (rows of data) often represent a small and non-representative sample.

This can dramatically impact the performance of machine learning algorithms fit on data with many input features, generally referred to as the “*curse of dimensionality*.”

Therefore, it is often desirable to reduce the number of input features. This reduces the number of dimensions of the feature space, hence the name “dimensionality reduction.”

A popular approach to dimensionality reduction is to use techniques from the field of linear algebra. This is often called “*feature projection*” and the algorithms used are referred to as “*projection methods*.”

Projection methods seek to reduce the number of dimensions in the feature space whilst also preserving the most important structure or relationships between the variables observed in the data.

“When dealing with high dimensional data, it is often useful to reduce the dimensionality by projecting the data to a lower dimensional subspace which captures the “essence” of the data. This is called dimensionality reduction.”

— Page 11, [Machine Learning: A Probabilistic Perspective, 2012.](#)

The resulting dataset, the projection, can then be used as input to train a machine learning model.

In essence, the original features no longer exist and new features are constructed from the available data that are not directly comparable to the original data, e.g. don't have column names.

Any new data that is fed to the model in the future when making predictions, such as test dataset and new datasets, must also be projected using the same technique.

Principal Component Analysis, or PCA, might be the most popular technique for dimensionality reduction.

“The most common approach to dimensionality reduction is called principal components analysis or PCA.”

— Page 11, [Machine Learning: A Probabilistic Perspective, 2012.](#)

It can be thought of as a projection method where data with m -columns (features) is projected into a subspace with m or fewer columns, whilst retaining the essence of the original data.

The PCA method can be described and implemented using the tools of linear algebra, specifically a matrix decomposition like an Eigendecomposition or SVD.

“PCA can be defined as the orthogonal projection of the data onto a lower dimensional linear space, known as the principal subspace, such that the variance of the projected data is maximized”

— Page 561, [Pattern Recognition and Machine Learning, 2006.](#)

PCA - How to do

Dimensionality Reduction, nói một cách đơn giản, là việc đi tìm một hàm số, hàm số này lấy đầu vào là một điểm dữ liệu ban đầu $\mathbf{x} \in \mathbb{R}^D$ với D rất lớn, và tạo ra một điểm dữ liệu mới $\mathbf{z} \in \mathbb{R}^K$ có số chiều $K < D$.

Tuy nhiên, nếu chúng ta có thể biểu diễn các vector dữ liệu ban đầu trong một hệ cơ sở mới mà trong hệ cơ sở mới đó, tầm quan trọng giữa các thành phần là khác nhau rõ rệt, thì chúng ta có thể bỏ qua những thành phần ít quan trọng nhất.

Lấy một ví dụ về việc có hai camera đặt dùng để chụp một con người, một camera đặt phía trước người và một camera đặt trên đầu. Rõ ràng là hình ảnh thu được từ camera đặt phía trước người mang nhiều thông tin hơn so với hình ảnh nhìn từ phía trên đầu. Vì vậy, bức ảnh chụp từ phía trên đầu có thể được bỏ qua mà không có quá nhiều thông tin về hình dáng của người đó bị mất.

PCA chính là phương pháp đi tìm một hệ cơ sở mới sao cho thông tin của dữ liệu chủ yếu tập trung ở một vài tọa độ, phần còn lại chỉ mang một lượng nhỏ thông tin. Và để cho đơn giản trong tính toán, PCA sẽ tìm một hệ trục chuẩn để làm cơ sở mới.

Giả sử hệ cơ sở trục chuẩn mới là \mathbf{U} và chúng ta muốn giữ lại K tọa độ trong hệ cơ sở mới này. Không mất tính tổng quát, giả sử đó là K thành phần đầu tiên

$$\begin{array}{c}
 \begin{array}{|c|} \hline N \\ \hline \end{array} \\
 \begin{array}{|c|} \hline D \\ \hline \end{array} \quad \mathbf{X}
 \end{array}
 =
 \begin{array}{|c|} \hline K \\ \hline \end{array}
 \begin{array}{|c|} \hline D-K \\ \hline \end{array}
 \begin{array}{|c|} \hline \mathbf{U}_K \\ \hline \end{array}
 \times
 \begin{array}{|c|} \hline N \\ \hline \end{array}
 \begin{array}{|c|} \hline K \\ \hline \end{array}
 \begin{array}{|c|} \hline \mathbf{Z} \\ \hline \end{array}
 \begin{array}{|c|} \hline D-K \\ \hline \end{array}
 \begin{array}{|c|} \hline \mathbf{Y} \\ \hline \end{array}$$

Original data An orthogonal matrix Coordinates in new basis

$$=
 \begin{array}{|c|} \hline K \\ \hline \end{array}
 \begin{array}{|c|} \hline D \\ \hline \end{array}
 \begin{array}{|c|} \hline \mathbf{U}_K \\ \hline \end{array}
 \times
 \begin{array}{|c|} \hline N \\ \hline \end{array}
 \begin{array}{|c|} \hline K \\ \hline \end{array}
 \begin{array}{|c|} \hline \mathbf{Z} \\ \hline \end{array}
 \begin{array}{|c|} \hline D \\ \hline \end{array}
 +
 \begin{array}{|c|} \hline \mathbf{U}_K \\ \hline \end{array}
 \times
 \begin{array}{|c|} \hline \mathbf{Y} \\ \hline \end{array}$$

Ý tưởng chính của PCA: Tìm một hệ trục chuẩn mới sao cho trong hệ này, các thành phần quan trọng nhất nằm trong K thành phần đầu tiên.

Các bước thực hiện PCA

1. Tính vector kỳ vọng của toàn bộ dữ liệu:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n$$

2. Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu:

$$\hat{\mathbf{x}}_n = \mathbf{x}_n - \bar{\mathbf{x}}$$

3. Tính ma trận hiệp phương sai

$$\mathbf{S} = \frac{1}{N} \hat{\mathbf{X}} \hat{\mathbf{X}}^T$$

4. Tính các trị riêng và vector riêng có norm bằng 1 của ma trận này, sắp xếp chúng theo thứ tự giảm dần của trị riêng.
5. Chọn K vector riêng ứng với K trị riêng lớn nhất để xây dựng ma trận \mathbf{U}_K có các cột tạo thành một hệ trục giao. K vectors này, còn được gọi là các thành phần chính, tạo thành một không gian con gần với phân bố của dữ liệu ban đầu đã chuẩn hoá.
6. Chiếu dữ liệu ban đầu đã chuẩn hoá $\hat{\mathbf{X}}$ xuống không gian con tìm được.
7. Dữ liệu mới chính là toạ độ của các điểm dữ liệu trên không gian mới

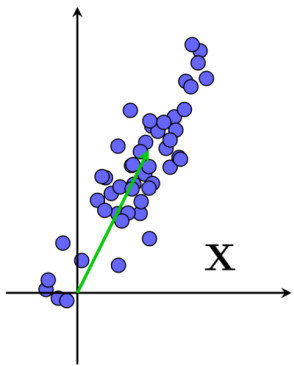
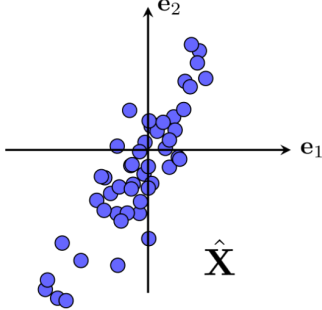
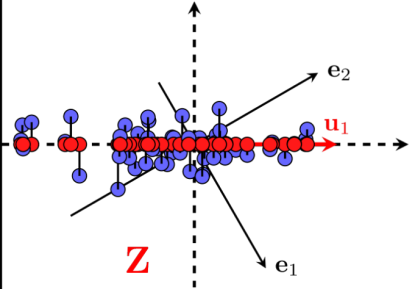
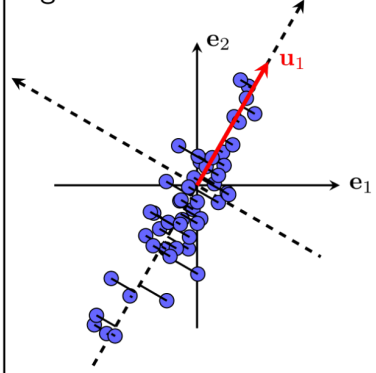
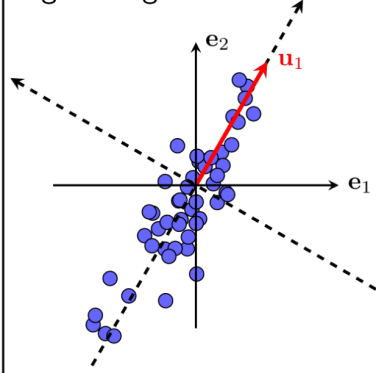
$$\mathbf{Z} = \mathbf{U}_K^T \hat{\mathbf{X}}$$

Dữ liệu ban đầu có thể tính được xấp xỉ theo dữ liệu mới như sau:

$$\mathbf{x} \approx \mathbf{U}_K \mathbf{Z} + \bar{\mathbf{x}}$$

Các bước thực hiện PCA có thể được xem trong Hình dưới đây:

PCA procedure

<p>1. Find mean vector</p>  <p>\mathbf{X}</p>	<p>2. Subtract mean</p>  <p>$\hat{\mathbf{X}}$</p>	<p>3. Compute covariance matrix: $\mathbf{S} = \frac{1}{N} \hat{\mathbf{X}} \hat{\mathbf{X}}^T$</p> <p>4. Compute eigenvalues and eigenvectors of \mathbf{S}: $(\lambda_1, \mathbf{u}_1), \dots, (\lambda_D, \mathbf{u}_D)$ Remember the orthonormality of \mathbf{u}_i.</p>
<p>7. Obtain projected points in low dimension.</p>  <p>\mathbf{Z}</p>	<p>6. Project data to selected eigenvectors.</p> 	<p>5. Pick K eigenvectors w. highest eigenvalues</p> 

References

1. [Principal Component Analysis for Visualization - MachineLearningMastery.com](#)
2. [How to Calculate Principal Component Analysis \(PCA\) from Scratch in Python - MachineLearningMastery.com](#)
3. [Principal Component Analysis for Dimensionality Reduction in Python - MachineLearningMastery.com](#)
4. [sklearn.decomposition.PCA](#)
5. [Bài 27: Principal Component Analysis \(phần 1/2\)| Machine Learning cơ bản \(machinelearningcoban.com\)](#)
6. [Bài 28: Principal Component Analysis \(phần 2/2\) | Machine Learning cơ bản \(machinelearningcoban.com\)](#)