Machine Learning: Dimension Reduction, Covariance and Principal Component Analysis (PCA)

Le Wang

Resources

This tutorial is drawn from many different sources, for example,

1. A.I. Wiki
2. Robert T. Collins at PSU
3. Learning Machines by Holger K. von Jouanne-Diedrich
4. Dan Ventura
5. Applications in Financial Economics
6. Principal Component Analysis using R by KURT HORNIK

Machine Learning: A Unsupervised Learning Technique for Dimension Reduction

Dimension Reduction: is prevalent in econometrics, statistics, and this modern so-called machine learning. In the presence of high-dimensional data (big data), it would be much easier to identify patterns when simplifying the data and reducing it to a lower dimensional data first.

The key is to **preserve important information** from the original data.

But do note most models are dimension reduction techniques as well. For example, linear regression

$$y = x'\beta + \epsilon$$

Examples:

1. **Machine Learning** images compression or facial recognition (eigenfaces) or voice recognition (eigenvoices)
2. **Economics** consumption behavior (expenditure patterns).
3. **Machine Learning and Economic Modelling** preprocess data for their models such as neural networks to improve the neural network's convergence speed and the overall quality of results.

What sounds complicated is really something we encounter every day: when we watch TV we see a 2D-projection of 3D-objects!

Look at my tea pot, how can you take a 2-D picture of it to best represent it?

3D Tea Pot

Lessons: there are many ways to represent teapot, which depends on how we rotate and scale it.

This concept naturally lends itself to linear algebra, and Pincipal Component Analysis is ONLY one technique that can be used.

Lessons: there are many ways to represent teapot, which depends on how we rotate and scale it.

This concept naturally lends itself to linear algebra, and Pincipal Component Analysis is ONLY one technique that can be used.

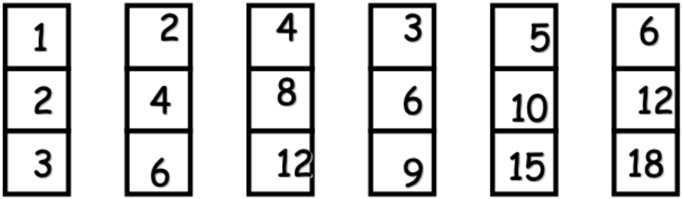
It effectively reduces a large set of correlated variables into a smaller set of **uncorrelated** variables, called **principal components** (which still contain most of the information).

PCA is a linear transformation that chooses a new coordinate system for the data set such that greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component), the second greatest variance on the second axis, and so on.

PCA can be used for reducing dimensionality by eliminating the later (or smaller) principal components.

PCA Toy Numerical Example

Consider the following 3D points



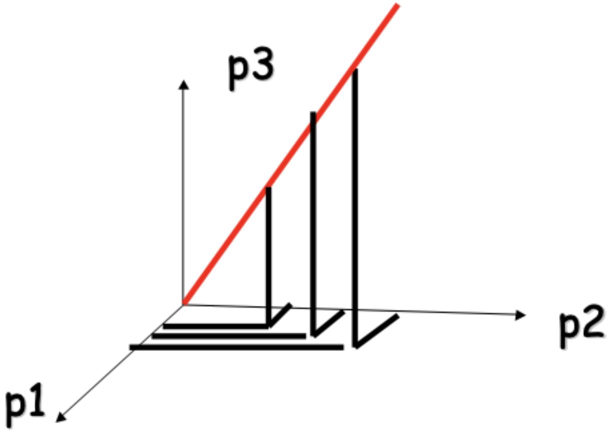
1	2	4	3	5	6
2	4	8	6	10	12
3	6	12	9	15	18

If each component is stored in a byte, we need $18 = 3 \times 6$ bytes

$$\begin{bmatrix}1\\2\\3\end{bmatrix} = 1 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}2\\4\\6\end{bmatrix} = 2 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}4\\8\\12\end{bmatrix} = 4 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

$$\begin{bmatrix}3\\6\\9\end{bmatrix} = 3 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}5\\10\\15\end{bmatrix} = 5 * \begin{bmatrix}1\\2\\3\end{bmatrix} \quad \begin{bmatrix}6\\12\\18\end{bmatrix} = 6 * \begin{bmatrix}1\\2\\3\end{bmatrix}$$

They can be stored using only 9 bytes (50 percent savings!): Store one point (3 bytes) + the multiplying constants (6 bytes)



We can just find out their corresponding projection points on the straight line. These points happen to be on the same straight line, so just the points on the line (in the same order as well).

What is the optimal way (for Mapping)?

It depends!

What is the optimal way (for Mapping)?

It depends!

PCA is just **one** way to do compress your data or to find a variable to summarize the important information among high-dimension variables.

Question: How does PCA find these components?

Question: How does PCA find these components?

Answer: By finding eigenvalues and eigenvectors of the variance-covariance matrix!

A Numerical Example with artificial data:

$$\left\{ \begin{pmatrix} -20 \\ -8 \end{pmatrix}, \begin{pmatrix} -10 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 10 \\ 1 \end{pmatrix}, \begin{pmatrix} 20 \\ 8 \end{pmatrix} \right\}$$

Step 1. Find the variance-covariance matrix first

$$\begin{pmatrix} 250 & 85 \\ 85 & 32.5 \end{pmatrix}$$

Step 2. Find Eigenvalues and Eigenvectors

$$\lambda_1 = 279.277398, \quad \lambda_2 = 3.222602$$

$$V = \begin{pmatrix} -0.95 & 0.33 \\ -0.33 & -0.95 \end{pmatrix}$$

Note that I am rounding up the values in the eigenvectors to ease illustration later.

Step 3. Choose the eigenvectors to Reduce dimensions

The eigenvectors correspond to principal components and the eigenvalues to the variance explained by the principal components.

Note that the first eigenvalue is much larger than the other, indicating significant difference in the variance in these two dimensions. We can just pick the first vector in V as our **Principal Component**.

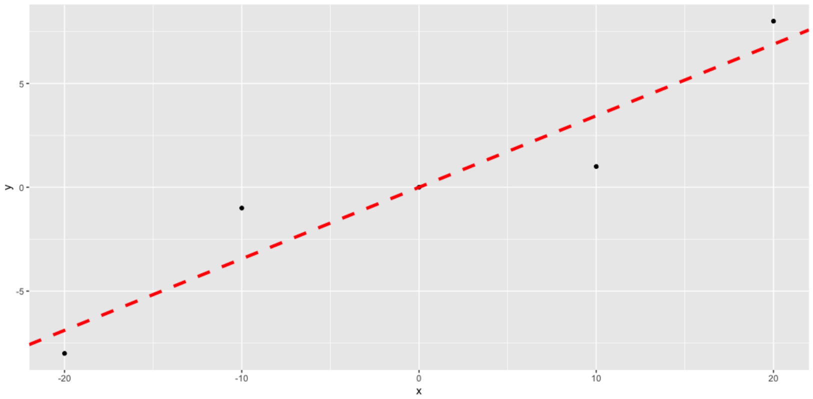
$$\begin{pmatrix} -0.95 & -0.33 \end{pmatrix} \begin{pmatrix} -20 \\ -8 \end{pmatrix} = (21.64)$$

$$\begin{pmatrix} -0.95 & -0.33 \end{pmatrix} \begin{pmatrix} -10 \\ -1 \end{pmatrix} = (9.83)$$

$$\begin{pmatrix} -0.95 & -0.33 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = (0)$$

$$\begin{pmatrix} -0.95 & -0.33 \end{pmatrix} \begin{pmatrix} 10 \\ 1 \end{pmatrix} = (-9.83)$$

$$\begin{pmatrix} -0.95 & -0.33 \end{pmatrix} \begin{pmatrix} 20 \\ 8 \end{pmatrix} = (-21.64)$$


5
0
-5
y
-20
-10
0
10
20
x

Application: Eigenfaces

A set of **eigenfaces** can be generated by performing a mathematical process called principal component analysis (PCA) on a large set of images depicting different human faces. Informally, eigenfaces can be considered a set of “standardized face ingredients”, derived from statistical analysis of many pictures of faces.

Any human face be considered to be a combination of these standard faces. For example, one’s face might be composed of the average face plus 10 percent from eigenface 1, 5 percent from eigenface 2, and even -3 from eigenface 3.

1. R example here
2. Another R example

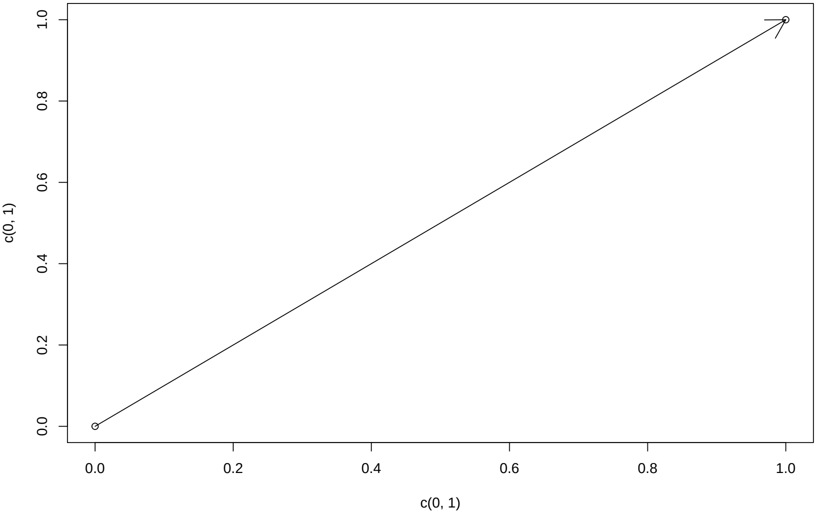
Mathematics behind this (Eigen-Stuff)

Eigenvalues and Eigenvectors

1. PCA
2. Google Pagerank
3. etc.

There are several things that one needs to understand.

First, any vector x is associated with a value of length and a direction.



Linear Transformation and Matrix Notice that for any $n \times n$ matrix A and a $n \times 1$ vector, x, the following operation

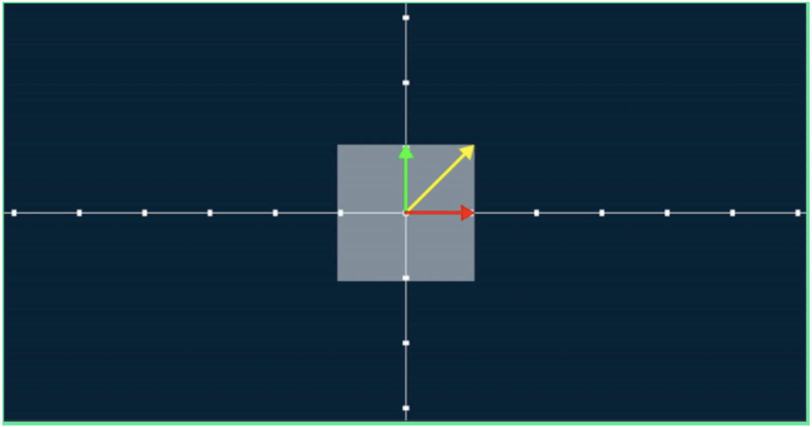
$$Ax$$

performs a linear transformation of x that "moves" it by changing its **direction** and scaling it.

Watch this video tutorial by 3Blue1Brown on linear transformations

Linear Transformation and Eigen-stuff

Eigen is a German word which means “own”, “proper” or “characteristic”. Eigenvectors and Eigenvalues are the characteristics of a particular matrix A. Lets visualize it first. This example is drawn from here.



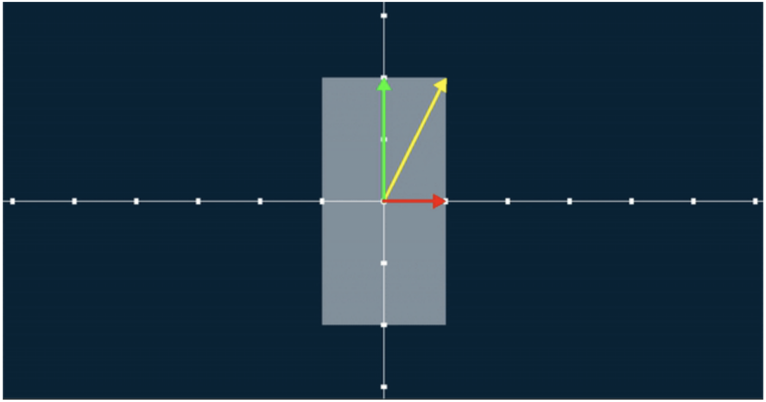

Scaling by a factor of 2 along y-axis

Figure 2: Example

Some linear transformation is applied to the three vectors (in other words, each vector was multiplied by a matrix A), the resultant vectors are displayed.

As we can see,

1. **red vector** does not change its direction nor the scale.
2. **gree vector** does not change its direction, but only the scale.
3. **yellow** changes both.

Both red and green vectors are called **eigenvectors** because their respective direction does not change in this linear transformation. Their corresponding scales are called **eigenvalues**.

Mathematically, it is formally given by

$$Ax = \lambda x$$

where x is a non-zero vector and λ is its corresponding eigenvalue.

The main thing is that the matrix is **square** and **symmetric**, which guarantees that the eigenvalues, λ_i, are real numbers.

Covariance matrices are also **positive semi-definite**, meaning that their eigenvalues are non-negative, $\lambda_i \geq 0$.

For the **covariance** or **correlation** matrix, the eigenvectors correspond to principal components and the eigenvalues to the variance explained by the principal components.

Principal component analysis of the correlation matrix provides an orthogonal basis for the space of the observed data: In this basis, the largest eigenvalues correspond to the principal components that are associated with most of the covariability among a number of observed data.

Spectral Decomposition:

$$A = VDV^{-1}$$

where V is the set of all **eigenvectors** and

$$D = \begin{pmatrix} \lambda_1, & 0, & \dots, & 0 \\ 0, & \lambda_2, & \dots, & 0 \\ \vdots, & \vdots, & \ddots, & 0 \\ 0, & 0, & \dots, & \lambda_k \end{pmatrix}$$

and $\lambda_1, \dots, \lambda_k$ are all the corresponding **eigenvalues**.

Note that for a non-square matrix, we need to apply singular value decomposition in the form

$$A = UDV'$$

In R,

```
svd <- svd(A)
svd$u%*%diag(svd$d)%*%t(svd$v)
```

Extension

Large Scale Eigenvalue decomposition