The Data Science Cycle

Feature Generation with High Information

DataLab

September 23, 2016

Outline

1. Introduction
   - What do we want?
2. Principal Component Analysis
   - Karhunen-Loeve Transform
   - PCA as a Linear Combination
   - Putting All Together

DataLab
Data Science Community

Outline

1. Introduction
   - What do we want?
2. Principal Component Analysis
   - Karhunen-Loeve Transform
   - PCA as a Linear Combination
   - Putting All Together

What do we want?

Given a set of measurements

The goal is to discover compact and informative representations of the obtained data.

Our Approach

We want to “squeeze” in a relatively small number of features.

Thus

Thus removing information redundancies.

What do we want?

Given a set of measurements

The goal is to discover compact and informative representations of the obtained data.

Our Approach

We want to "squeeze" in a relatively small number of features.

Thus

Thus removing information redundancies.

What do we want?

Given a set of measurements

The goal is to discover compact and informative representations of the obtained data.

Our Approach

We want to "squeeze" in a relatively small number of features.

Thus

Thus removing information redundancies.

Outline

1. Introduction
   - What do we want?
2. Principal Component Analysis
   - Karhunen-Loeve Transform
   - PCA as a Linear Combination
   - Putting All Together

DataLab
Data Science Community

Also Known as Karhunen-Loeve Transform

Setup

- Consider a data set of observations $\{\boldsymbol{x}_n\}$ with $n = 1, 2, ..., N$ and $\boldsymbol{x_n} \in R^d$.

Goal

Project data onto space with dimensionality $m < d$ (We assume m is given)

Also Known as Karhunen-Loeve Transform

Setup

- Consider a data set of observations $\{\boldsymbol{x}_n\}$ with $n = 1, 2, ..., N$ and $\boldsymbol{x_n} \in R^d$.

Goal

Project data onto space with dimensionality $m < d$ (We assume m is given)

What PCA is Asking?

Question

Is there another basis, which is a linear combination of the original basis, that best re-expresses our data set?

Therefore

PCA assumes linearity by stating that the data set even characterizes the system!!!

Therefore

PCA relies on **the superposition principal of linearity** to believe that the data provides an ability to interpolate between the individual data points!!!

What PCA is Asking?

Question

Is there another basis, which is a linear combination of the original basis, that best re-expresses our data set?

Therefore

PCA assumes linearity by stating that the data set even characterizes the system!!!

Therefore

PCA relies on **the superposition principal of linearity** to believe that the data provides an ability to interpolate between the individual data points!!!

What PCA is Asking?

Question

Is there another basis, which is a linear combination of the original basis, that best re-expresses our data set?

Therefore

PCA assumes linearity by stating that the data set even characterizes the system!!!

Therefore

PCA relies on **the superposition principal of linearity** to believe that the data provides an ability to interpolate between the individual data points!!!

Outline

1 Introduction
- What do we want?

2 Principal Component Analysis
- Karhunen-Loeve Transform
- PCA as a Linear Combination
- Putting All Together

DataLab
Data Science Community

PCA as a Linear Combination

Thus

PCA is now limited to re-expressing the data as a linear combination of its basis vectors.

Then, given X and Y be $m \times n$ matrices related by

$$PX = Y$$

Where

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix}, \quad X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}, \quad Y = \begin{bmatrix} p_1 \cdot x_1 & \cdots & p_1 \cdot x_n \\ \vdots & \ddots & \vdots \\ p_m \cdot x_1 & \cdots & p_m \cdot x_n \end{bmatrix}$$

DataLab
Data Science Community

PCA as a Linear Combination

Thus

PCA is now limited to re-expressing the data as a linear combination of its basis vectors.

Then, given X and Y be $m \times n$ matrices related by

$$PX = Y$$

Where

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{bmatrix}, \quad X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}, \quad Y = \begin{bmatrix} p_1 \cdot x_1 & \cdots & p_1 \cdot x_n \\ \vdots & \ddots & \vdots \\ p_m \cdot x_1 & \cdots & p_m \cdot x_n \end{bmatrix}$$

PCA as a Linear Combination

Thus

PCA is now limited to re-expressing the data as a linear combination of its basis vectors.

Then, given X and Y be $m \times n$ matrices related by

$$PX = Y$$

Where

$$P = \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_m \end{bmatrix}, \quad X = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_n \end{bmatrix}, \quad Y = \begin{bmatrix} \boldsymbol{p}_1 \cdot \boldsymbol{x}_1 & \cdots & \boldsymbol{p}_1 \cdot \boldsymbol{x}_n \\ \vdots & \ddots & \vdots \\ \boldsymbol{p}_m \cdot \boldsymbol{x}_1 & \cdots & \boldsymbol{p}_m \cdot \boldsymbol{x}_n \end{bmatrix}$$

Therefore

We have two questions

- What is the best way to "re-express" X?
- What is a good choice of basis P?

The Goal

Decipher Garbled Data!!!

How

Dealing with noise and redundancy!!!

Therefore

We have two questions

- What is the best way to "re-express" X?
- What is a good choice of basis P?

The Goal

Decipher Garbled Data!!!

How

Dealing with noise and redundancy!!!

Therefore

We have two questions

- What is the best way to “re-express” X?
- What is a good choice of basis P?

The Goal

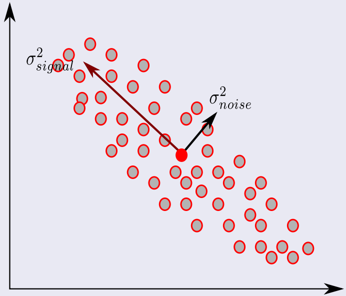
Decipher Garbled Data!!!

How

Dealing with noise and redundancy!!!

Assume the following

Imagine the following

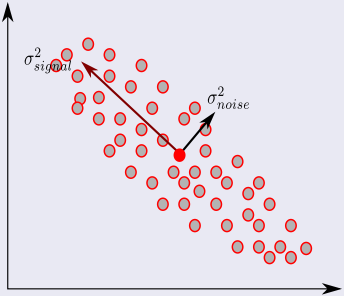


Thus, we have the following measure

$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}$$

Assume the following

Imagine the following



Thus, we have the following measure

$$SNR = \frac{\sigma^2_{signal}}{\sigma^2_{noise}}$$

DataLab
Data Science Community

What SNR is telling us?

What do we have

- $SNR \gg 1$ High Precision Data.
- $SNR = 1$ Represent Data Highly contaminated by noise.

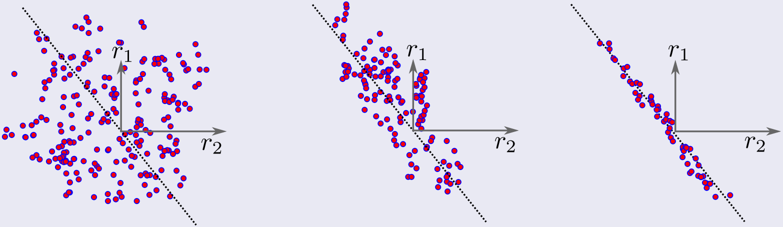
Here

We will assume that our data does not have that much noise

Then PCA tries to find the directions where that noise does not affect the observations!!!

Additionally, we have the following phenomena

Here is the problem



Outline

1. Introduction
   - What do we want?
2. Principal Component Analysis
   - Karhunen-Loeve Transform
   - PCA as a Linear Combination
   - Putting All Together

DataLab
Data Science Community

Then, we do the following

Given two sets of simultaneous measurements with zero mean

$$X = \{x_1, x_2, ..., x_n\}, Y = \{y_1, y_2, ..., y_n\}$$

Therefore

$$\sigma_X^2 = E[x_i x_i], \sigma_X^2 = E[y_i y_i]$$

In the general case

$$\sigma_{XY}^2 = E[x_i y_i]$$

Then, we do the following

Given two sets of simultaneous measurements with zero mean

$$X = \{x_1, x_2, ..., x_n\}, Y = \{y_1, y_2, ..., y_n\}$$

Therefore

$$\sigma_X^2 = E[x_i x_i], \sigma_X^2 = E[y_i y_i]$$

In the general case

$$\sigma_{XY}^2 = E[x_i y_i]$$

Then, we do the following

Given two sets of simultaneous measurements with zero mean

$$X = \{x_1, x_2, ..., x_n\}, Y = \{y_1, y_2, ..., y_n\}$$

Therefore

$$\sigma_X^2 = E[x_i x_i], \sigma_X^2 = E[y_i y_i]$$

In the general case

$$\sigma_{XY}^2 = E[x_i y_i]$$

DataLab
Data Science Community

Variance in One Dimension

Remember the Sample Variance

$$VAR(X) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(x_i - \overline{x})}{N-1} \tag{1}$$

You can do the same in the case of two variables X and Y

$$COV(X,Y) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{N-1} \tag{2}$$

Variance in One Dimension

Remember the Sample Variance

$$VAR(X) = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(x_i - \overline{x})}{N-1} \tag{1}$$

You can do the same in the case of two variables X and Y

$$COV(X,Y) = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{N-1} \tag{2}$$

Thus

Two important facts about the covariance

- $\sigma^2_{XY} = 0$ if and only if A and B are entirely uncorrelated.
- $\sigma^2_{XY} = \sigma^2_X$ if $X = Y$.

Now, we can express the covariance as

$$\sigma^2_{XY} = \frac{1}{N-1} XY^T$$

Thus

Two important facts about the covariance

- $\sigma^2_{XY} = 0$ if and only if A and B are entirely uncorrelated.
- $\sigma^2_{XY} = \sigma^2_X$ if $X = Y$.

Now, we can express the covariance as

$$\sigma^2_{XY} = \frac{1}{N-1} XY^T$$

DataLab
Data Science Community

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{3}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{4}$$

Build new data

$$X = [\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}}] \tag{5}$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{3}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{4}$$

Build new data

$$X = [\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}}] \tag{5}$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{3}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{4}$$

Build new data

$$X = \left[\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}}\right] \tag{5}$$

DataLab
Data Science Community

Build the Sample Covariance

The Multivariate Covariance Matrix

$$S = \frac{1}{N-1} X X^T \tag{6}$$

Properties

1. The ijth value of S is equivalent to σ_{ij}^2.
2. The iith value of S is equivalent to σ_{ii}^2.
3. What else? Look at a plane Center and Rotating!!!

DataLab
Data Science Community

Build the Sample Covariance

The Multivariate Covariance Matrix

$$S = \frac{1}{N-1} X X^T \tag{6}$$

Properties

1. The ijth value of S is equivalent to σ_{ij}^2.
2. The iith value of S is equivalent to σ_{ii}^2.
3. What else? Look at a plane Center and Rotating!!!

DataLab
Data Science Community

Using S to Project Data

Project the data

We want to project the data to a line...

For this we use a u_1

with $u_1^T u_1 = 1$

Using S to Project Data

Project the data

We want to project the data to a line...

For this we use a $\boldsymbol{u}_1$

with $\boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$

DataLab
Data Science Community

Thus we have

Variance of the projected data

$$\frac{1}{N-1}\sum_{i=1}^{N}\left[\boldsymbol{u}_1\boldsymbol{x}_i-\boldsymbol{u}_1\overline{\boldsymbol{x}}\right]=\boldsymbol{u}_1^T S\boldsymbol{u}_1 \tag{7}$$

Use Lagrange Multipliers to Maximize

$$\boldsymbol{u}_1^T S\boldsymbol{u}_1+\lambda_1\left(1-\boldsymbol{u}_1^T\boldsymbol{u}_1\right) \tag{8}$$

Thus we have

Variance of the projected data

$$\frac{1}{N-1}\sum_{i=1}^{N}\left[\boldsymbol{u}_1\boldsymbol{x}_i - \boldsymbol{u}_1\overline{\boldsymbol{x}}\right] = \boldsymbol{u}_1^T S \boldsymbol{u}_1 \tag{7}$$

Use Lagrange Multipliers to Maximize

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 + \lambda_1\left(1 - \boldsymbol{u}_1^T\boldsymbol{u}_1\right) \tag{8}$$

DataLab
Data Science Community

Derive by $\boldsymbol{u}_1$

We get

$$S\boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \quad (9)$$

Then

$\boldsymbol{u}_1$ is an eigenvector of S.

If we left-multiply by $\boldsymbol{u}_1$

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \quad (10)$$

DataLab
Data Science Community

Derive by $\boldsymbol{u}_1$

We get

$$S\boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \tag{9}$$

Then

$\boldsymbol{u}_1$ is an eigenvector of S.

If we left-multiply by $\boldsymbol{u}_1$

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{10}$$

DataLab
Data Science Community

Derive by $\boldsymbol{u}_1$

We get

$$S\boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \tag{9}$$

Then

$\boldsymbol{u}_1$ is an eigenvector of S.

If we left-multiply by $\boldsymbol{u}_1$

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{10}$$

DataLab
Data Science Community

Thus

Variance will be the maximum when

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{11}$$

is set to the largest eigenvalue. Also know as the First Principal Component

By Induction

It is possible for M-dimensional space to define M eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M$ of the data covariance S corresponding to $\lambda_1, \lambda_2, ..., \lambda_M$ that maximize the variance of the projected data.

Computational Cost

1. Full eigenvector decomposition $O\left(d^3\right)$
2. Power Method $O\left(Md^2\right)$ "Golub and Van Loan, 1996)"
3. Use the Expectation Maximization Algorithm

Thus

Variance will be the maximum when

$$u_1^T S u_1 = \lambda_1 \tag{11}$$

is set to the largest eigenvalue. Also know as the First Principal Component

By Induction

It is possible for M-dimensional space to define M eigenvectors $u_1, u_2, ..., u_M$ of the data covariance S corresponding to $\lambda_1, \lambda_2, ..., \lambda_M$ that maximize the variance of the projected data.

Computational Cost

1. Full eigenvector decomposition $O\left(d^3\right)$
2. Power Method $O\left(Md^2\right)$ "Golub and Van Loan, 1996)"
3. Use the Expectation Maximization Algorithm

Thus

Variance will be the maximum when

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \quad (11)$$

is set to the largest eigenvalue. Also know as the First Principal Component

By Induction

It is possible for M-dimensional space to define M eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M$ of the data covariance S corresponding to $\lambda_1, \lambda_2, ..., \lambda_M$ that maximize the variance of the projected data.

Computational Cost

1. Full eigenvector decomposition $O\left(d^3\right)$
2. Power Method $O\left(Md^2\right)$ "Golub and Van Loan, 1996)"
3. Use the Expectation Maximization Algorithm

In Our Case, we will use

The following instruction

- np.linalg.egh$\left(\widehat{\Sigma}\right)$

This returns

The eigenvalues and the eigenvectors (The new Base!!!)

DataLab
Data Science Community

In Our Case, we will use

The following instruction

- np.linalg.egh$\left(\widehat{\Sigma}\right)$

This returns

The eigenvalues and the eigenvectors (The new Base!!!)

DataLab
Data Science Community

Thus

Given a data set X

We need to implement the mean per features

- Xmean = X - np.mean(X,axis = 0)

Then creating the Covariance

- Cov = DataMean.T*DataMean
- n1, n2 = Data.shape
- Cov = (1/float(n1-1))*Cov

Then, we obtain the desired values

- Eigenvaluesc, Eigenvectorsc = np.linalg.eigh(Cov)
- idx = Eigenvaluesc.argsort()[::-1]
- Eigenvaluesc = Eigenvaluesc[idx]
- Eigenvectorsc = Eigenvectorsc [:,idx]

Thus

Given a data set X

We need to implement the mean per features

- Xmean = X - np.mean(X,axis = 0)

Then creating the Covariance

- Cov = DataMean.T*DataMean
- n1, n2 = Data.shape
- Cov = (1/float(n1-1))*Cov

Then, we obtain the desired values

- Eigenvaluesc, Eigenvectorsc = np.linalg.eigh(Cov)
- idx = Eigenvaluesc.argsort()[::-1]
- Eigenvaluesc = Eigenvaluesc[idx]
- Eigenvectorsc = Eigenvectorsc [:,idx]

Thus

Given a data set X

We need to implement the mean per features

- Xmean = X - np.mean(X,axis = 0)

Then creating the Covariance

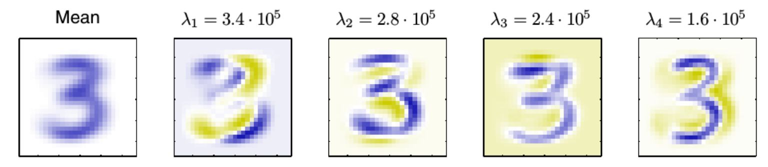
- Cov = DataMean.T*DataMean
- n1, n2 = Data.shape
- Cov = (1/float(n1-1))*Cov

Then, we obtain the desired values

- Eigenvaluesc, Eigenvectorsc = np.linalg.eigh(Cov)
- idx = Eigenvaluesc.argsort()[::-1]
- Eigenvaluesc = Eigenvaluesc[idx]
- Eigenvectorsc = Eigenvectorsc [:,idx]

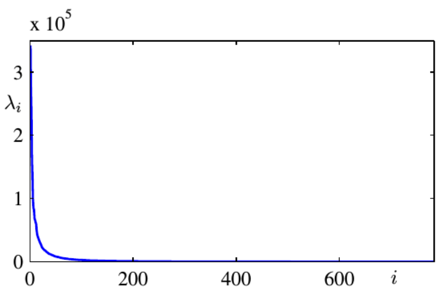
Example

From Bishop



Example

From Bishop



Example

From Bishop