The Data Science Cycle

Feature Selection

DataLab

September 21, 2016

Outline

DataLab
Data Science Community

Outline

Loading Data

We can use Pandas for this!!!

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import chi2


def ReturnDataFrame(self, path):
        return pd.read_csv(path, sep=',',
                           skipinitialspace=True)

# Load CVS
Path1 = 'SomePath'
DataMatrix = ReturnDataFrame(Path1)

# Transform to an NP Array
Data = DataMatrix.as_matrix()

Data = Data.astype(float)
```

Some properties of the new numpy matrix

We have

$$
\begin{array}{cc}
 & \overbrace{\hphantom{x_{12} \quad \cdots}}^{Features} \\
Samples \left\{ \right. & \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nd} \end{pmatrix}
\end{array}
$$

DataLab
Data Science Community

Outline

DataLab
Data Science Community

What is this?

Main Question

"Given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as much as possible of their class discriminatory information? "

What is this?

Main Question

"Given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as much as possible of their class discriminatory information? "

Why is important?

1. If we selected features with little discrimination power, the subsequent design of a classifier would lead to poor performance.
2. if information-rich features are selected, the design of the classifier can be greatly simplified.

- Large between class distance.
- Small within class variance.

What is this?

Main Question

"Given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as much as possible of their class discriminatory information? "

Why is important?

1. If we selected features with little discrimination power, the subsequent design of a classifier would lead to poor performance.
2. if information-rich features are selected, the design of the classifier can be greatly simplified.

Therefore

We want features that lead to

1. Large between-class distance.
2. Small within-class variance.

What is this?

Main Question

"Given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as much as possible of their class discriminatory information? "

Why is important?

1. If we selected features with little discrimination power, the subsequent design of a classifier would lead to poor performance.
2. if information-rich features are selected, the design of the classifier can be greatly simplified.

Therefore

We want features that lead to

1. Large between-class distance.
2. Small within-class variance.

What is this?

Main Question

"Given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as much as possible of their class discriminatory information? "

Why is important?

1. If we selected features with little discrimination power, the subsequent design of a classifier would lead to poor performance.
2. if information-rich features are selected, the design of the classifier can be greatly simplified.

Therefore

We want features that lead to

1. Large between-class distance.
2. Small within-class variance.

What is this?

Main Question

"Given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as much as possible of their class discriminatory information? "

Why is important?

1. If we selected features with little discrimination power, the subsequent design of a classifier would lead to poor performance.
2. if information-rich features are selected, the design of the classifier can be greatly simplified.

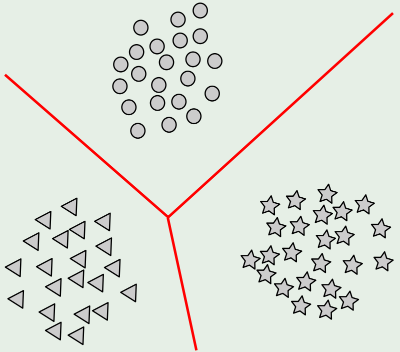
Therefore

We want features that lead to

1. Large between-class distance.
2. Small within-class variance.

Then

Basically, we want nice separated and dense clusters!!!



Outline

However, Before That...

It is necessary to do the following

1. Outliers removal.
2. Data normalization.
3. Deal with missing data.

However, Before That...

It is necessary to do the following

1. Outliers removal.
2. Data normalization.
3. Deal with missing data.

Actually

PREPROCESSING!!!

However, Before That...

It is necessary to do the following

1. Outliers removal.
2. Data normalization.
3. Deal with missing data.

Actually

PREPROCESSING!!!

However, Before That...

It is necessary to do the following

1. Outliers removal.
2. Data normalization.
3. Deal with missing data.

Actually

PREPROCESSING!!!

Outline

DataLab
Data Science Community

Outliers

Definition

An outlier is defined as a point that lies very far from the mean of the corresponding random variable.

Note: We use the standard deviation

Outliers

Definition

An outlier is defined as a point that lies very far from the mean of the corresponding random variable.

Note: We use the standard deviation

Example

For a normally distributed random variable

1. A distance of two times the standard deviation covers 95% of the points.
2. A distance of three times the standard deviation covers 99% of the points.

DataLab
Data Science Community

Outliers

Definition

An outlier is defined as a point that lies very far from the mean of the corresponding random variable.

Note: We use the standard deviation

Example

For a normally distributed random variable

1. A distance of two times the standard deviation covers 95% of the points.
2. A distance of three times the standard deviation covers 99% of the points.

DataLab
Data Science Community

Outliers

Definition

An outlier is defined as a point that lies very far from the mean of the corresponding random variable.

Note: We use the standard deviation

Example

For a normally distributed random variable

1. A distance of two times the standard deviation covers 95% of the points.
2. A distance of three times the standard deviation covers 99% of the points.

DataLab
Data Science Community

Outliers

Definition

An outlier is defined as a point that lies very far from the mean of the corresponding random variable.

Note: We use the standard deviation

Example

For a normally distributed random variable

1. A distance of two times the standard deviation covers 95% of the points.
2. A distance of three times the standard deviation covers 99% of the points.

Outliers

Note

- Points with values very different from the mean value produce large errors during training and may have disastrous effects.
- These effects are even worse when the outliers, and they are the result of noisy measurement.

Outliers

Note

- Points with values very different from the mean value produce large errors during training and may have disastrous effects.
- These effects are even worse when the outliers, and they are the result of noisy measurement.

DataLab
Data Science Community

Outlier Removal

Important

Then removing outliers is the biggest importance.

Outlier Removal

Important

Then removing outliers is the biggest importance.

Therefore

You can do the following

- If you have a small number $\Rightarrow$ discard them!!!

Outlier Removal

Important

Then removing outliers is the biggest importance.

Therefore

You can do the following

- If you have a small number $\Rightarrow$ discard them!!!

Outlier Removal

Important

Then removing outliers is the biggest importance.

Therefore

You can do the following

- If you have a small number $\Rightarrow$ discard them!!!

DataLab
Data Science Community

Outline

DataLab
Data Science Community

We can do the following

Algorithm

Input: An $N \times d$ data set $Data$

Output: Candidate Outliers

1. Calculate the sample mean μ and sample covariance matrix Σ.
2. Let M be $N \times 1$ vector consisting of square of the Mahalonobis distance to μ.
3. Find points O in M whose values are greater than

$$\chi^2_d(0.05)$$

4. Return O.

We can do the following

Algorithm

Input: An $N \times d$ data set $Data$

Output: Candidate Outliers

1. Calculate the sample mean μ and sample covariance matrix Σ.
2. Let M be $N \times 1$ vector consisting of square of the Mahalonobis distance to μ.
3. Find points O in M whose values are greater than

$$\chi_d^2(0.05)$$

4. Return O.

We can do the following

Algorithm

Input: An $N \times d$ data set $Data$

Output: Candidate Outliers

1. Calculate the sample mean μ and sample covariance matrix Σ.
2. Let M be $N \times 1$ vector consisting of square of the Mahalonobis distance to μ.
3. Find points O in M whose values are greater than

$$\chi_d^2(0.05)$$

4. Return O.

We can do the following

Algorithm

Input: An $N \times d$ data set $Data$

Output: Candidate Outliers

1. Calculate the sample mean μ and sample covariance matrix Σ.
2. Let M be $N \times 1$ vector consisting of square of the Mahalonobis distance to μ.
3. Find points O in M whose values are greater than

$$\chi^2_d(0.05)$$

4. Return O.

We can do the following

Algorithm

Input: An $N \times d$ data set $Data$

Output: Candidate Outliers

1. Calculate the sample mean μ and sample covariance matrix Σ.
2. Let M be $N \times 1$ vector consisting of square of the Mahalonobis distance to μ.
3. Find points O in M whose values are greater than

$$\chi^2_d(0.05)$$

4. Return O.

We can do the following

Algorithm

Input: An $N \times d$ data set $Data$

Output: Candidate Outliers

1. Calculate the sample mean μ and sample covariance matrix Σ.
2. Let M be $N \times 1$ vector consisting of square of the Mahalonobis distance to μ.
3. Find points O in M whose values are greater than

$$\chi^2_d\,(0.05)$$

4. Return O.

DataLab
Data Science Community

How?

Get the Sample Mean per feature k

$$\boldsymbol{m}_i = \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{x}_{ki}$$

Get the Sample Variance per feature k

$$v_i = \frac{1}{N-1}\sum_{k=1}^{N} \left(\boldsymbol{x}_{ki} - \boldsymbol{m}_i\right)\left(\boldsymbol{x}_{ki} - \boldsymbol{m}_i\right)^T$$

How?

Get the Sample Mean per feature k

$$\boldsymbol{m}_i = \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{x}_{ki}$$

Get the Sample Variance per feature k

$$v_i = \frac{1}{N-1}\sum_{k=1}^{N}\left(\boldsymbol{x}_{ki}-\boldsymbol{m}_i\right)\left(\boldsymbol{x}_{ki}-\boldsymbol{m}_i\right)^T$$

DataLab
Data Science Community

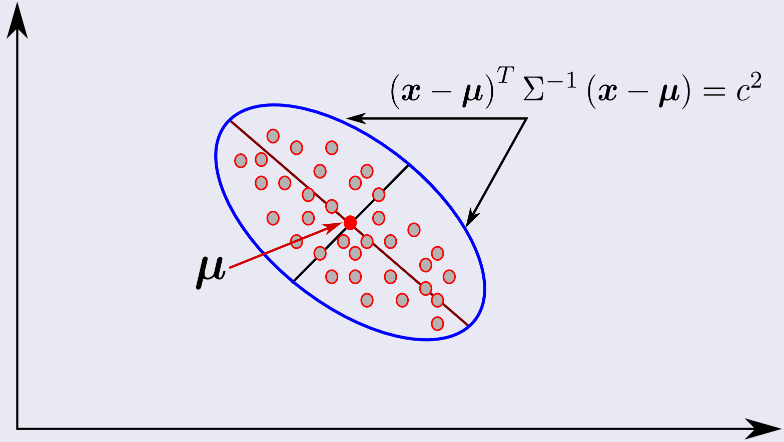
Mahalonobis Distance

We have

$$M(\boldsymbol{x}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})}$$

Why the line 3?

Setting $M(\boldsymbol{x})$ to a constant c defines a multidimensional ellipsoid with centroid at $\boldsymbol{\mu}$



As Johnson and Wichern (2007, p. 155, Eq. 4-8) state

The solid ellipsoid of $\boldsymbol{x}$ vectors satisfying

$$(\boldsymbol{x} - \boldsymbol{\mu})^T \, \Sigma^{-1} \, (\boldsymbol{x} - \boldsymbol{\mu}) \leq \chi^2_d \, (\alpha)$$

has a probability $1 - \alpha$.

DataLab
Data Science Community

Algorithm

The Partial Code

```
def OutlierRemoval(self, Data):
    SampleMean  = Data.mean(1)
    SampleCov   = Data - SampleMean
    SampleCov   = np.cov(SampleCov.T)
    Mahalonobis = (Data - SampleMean)*
                          np.inv(SampleCov)*
                          ((Data - SampleMean).T)

    # Something else here
    # Here you can use chi2.isf(\alpha,dim)
```

DataLab
Data Science Community

Outline

DataLab
Data Science Community

Data Normalization

In the real world

In many practical situations a designer is confronted with features whose values lie within different dynamic ranges.

For Example

We can have two features with the following ranges

$$
\begin{aligned}
x_i &\in [0, 100, 000] \\
x_j &\in [0, 0.5]
\end{aligned}
$$

Thus

Many classification machines will be swamped by the first feature!!!

Data Normalization

In the real world

In many practical situations a designer is confronted with features whose values lie within different dynamic ranges.

For Example

We can have two features with the following ranges

$$
\begin{aligned}
x_i &\in [0, 100, 000] \\
x_j &\in [0, 0.5]
\end{aligned}
$$

Thus

Many classification machines will be swamped by the first feature!!!

Data Normalization

In the real world

In many practical situations a designer is confronted with features whose values lie within different dynamic ranges.

For Example

We can have two features with the following ranges

$$\begin{aligned} x_i &\in [0, 100, 000] \\ x_j &\in [0, 0.5] \end{aligned}$$

Thus

Many classification machines will be swamped by the first feature!!!

Data Normalization

We have the following situation

Features with large values may have a larger influence in the cost function than features with small values.

Thus!!!

This does not necessarily reflect their respective significance in the design of the classifier.

Data Normalization

We have the following situation

Features with large values may have a larger influence in the cost function than features with small values.

Thus!!!

This does not necessarily reflect their respective significance in the design of the classifier.

DataLab
Data Science Community

Data Normalization

We have the following situation

Features with large values may have a larger influence in the cost function than features with small values.

Thus!!!

This does not necessarily reflect their respective significance in the design of the classifier.

Naive Normalization

Be Naive

For each feature $i = 1, ..., d$ obtain the $\max_i$ and the $\min_i$ such that

$$\hat{x}_{ik} = \frac{x_{ik} - \min_i}{\max_i - \min_i} \tag{1}$$

Problem

This simple normalization will send everything to a unitary sphere thus loosing data resolution!!!

Naive Normalization

Be Naive

For each feature $i = 1, ..., d$ obtain the $\max_i$ and the $\min_i$ such that

$$\hat{x}_{ik} = \frac{x_{ik} - \min_i}{\max_i - \min_i} \tag{1}$$

Problem

This simple normalization will send everything to a unitary sphere thus loosing data resolution!!!

DataLab
Data Science Community

Gaussian Scaling

Use the idea of

Everything is Gaussian...

Gaussian Scaling

Use the idea of

Everything is Gaussian...

Thus

For each feature set...

1. $\overline{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{ik},\ \ k = 1, 2, ..., d$
2. $\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_{ik} - \overline{x}_k\right)^2,\ \ k = 1, 2, ..., d$

Gaussian Scaling

Use the idea of

Everything is Gaussian...

Thus

For each feature set...

1. $\overline{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{ik}, \; k = 1, 2, ..., d$
2. $\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_{ik} - \overline{x}_k\right)^2, \; k = 1, 2, ..., d$

Thus

$$\hat{x}_{ik} = \frac{x_{ik} - \overline{x}_k}{\sigma} \tag{2}$$

Gaussian Scaling

Use the idea of

Everything is Gaussian...

Thus

For each feature set...

1. $\overline{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{ik},\ \ k = 1, 2, ..., d$
2. $\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N} \left(x_{ik} - \overline{x}_k\right)^2,\ \ k = 1, 2, ..., d$

Thus

$$\hat{x}_{ik} = \frac{x_{ik} - \overline{x}_k}{\sigma} \tag{2}$$

DataLab
Data Science Community

Gaussian Scaling

Use the idea of

Everything is Gaussian...

Thus

For each feature set...

1. $\overline{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{ik},\ \ k = 1, 2, ..., d$
2. $\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^{N} \left(x_{ik} - \overline{x}_k\right)^2,\ \ k = 1, 2, ..., d$

Thus

$$\hat{x}_{ik} = \frac{x_{ik} - \overline{x}_k}{\sigma} \tag{2}$$

DataLab
Data Science Community

For Example

We have

```python
def GaussianScaling(self, Data):
    SampleMean = np.mean(Data, axis = 0)
    SampleStd  = np.std(Data, axis = 0)

    return (Data-SampleMean)/SampleStd
```

DataLab
Data Science Community

Outline

DataLab
Data Science Community

Missing Data

This can happen

In practice, certain features may be missing from some feature vectors.

Missing Data

This can happen

In practice, certain features may be missing from some feature vectors.

Examples where this happens

1. Social sciences - incomplete surveys.
2. Remote sensing - sensors go off-line.
3. etc.

Missing Data

This can happen

In practice, certain features may be missing from some feature vectors.

Examples where this happens

1. Social sciences - incomplete surveys.
2. Remote sensing - sensors go off-line.
3. etc.

Note

Completing the missing values in a set of data is also known as imputation.

DataLab
Data Science Community

Missing Data

This can happen

In practice, certain features may be missing from some feature vectors.

Examples where this happens

1. Social sciences - incomplete surveys.
2. Remote sensing - sensors go off-line.
3. etc.

Note

Completing the missing values in a set of data is also known as imputation.

DataLab
Data Science Community

Missing Data

This can happen

In practice, certain features may be missing from some feature vectors.

Examples where this happens

1. Social sciences - incomplete surveys.
2. Remote sensing - sensors go off-line.
3. etc.

Note

Completing the missing values in a set of data is also known as imputation.

DataLab
Data Science Community

Some traditional techniques to solve this problem

Use zeros and risked it!!!

The idea is not to add anything to the features

The sample mean/unconditional mean

Does not matter what distribution you have use the sample mean

$$\overline{x}_i = \frac{1}{N}\sum_{k=1}^{N} x_{ik} \tag{3}$$

Find the distribution of your data

Use the mean from that distribution. For example, if you have a beta distribution

$$\overline{x}_i = \frac{\alpha}{\alpha+\beta} \tag{4}$$

Some traditional techniques to solve this problem

Use zeros and risked it!!!

The idea is not to add anything to the features

The sample mean/unconditional mean

Does not matter what distribution you have use the sample mean

$$\overline{x}_i = \frac{1}{N} \sum_{k=1}^{N} x_{ik} \tag{3}$$

Find the distribution of your data

Use the mean from that distribution. For example, if you have a beta distribution

$$\overline{x}_i = \frac{\alpha}{\alpha + \beta} \tag{4}$$

Some traditional techniques to solve this problem

Use zeros and risked it!!!

The idea is not to add anything to the features

The sample mean/unconditional mean

Does not matter what distribution you have use the sample mean

$$\overline{x}_i = \frac{1}{N} \sum_{k=1}^{N} x_{ik} \tag{3}$$

Find the distribution of your data

Use the mean from that distribution. For example, if you have a beta distribution

$$\overline{x}_i = \frac{\alpha}{\alpha + \beta} \tag{4}$$

The MOST traditional

Drop it

- Remove that data
  - Still you need to have a lot of data to have this luxury

Outline

THE PEAKING PHENOMENON

Remember

Normally, to design a classifier with good generalization performance, we want the number of sample N to be larger than the number of features d.

What?

The intuition, the larger the number of samples vs the number of features, the smaller the error P_e

THE PEAKING PHENOMENON

Remember

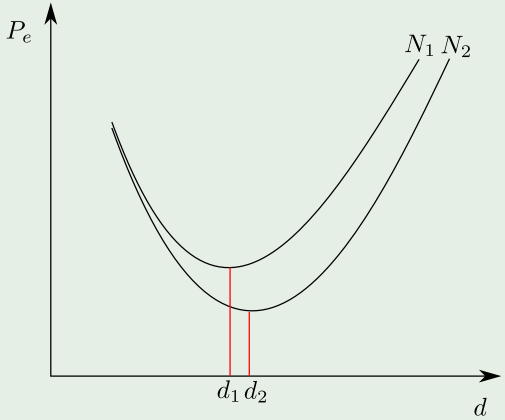
Normally, to design a classifier with good generalization performance, we want the number of sample N to be larger than the number of features d.

What?

The intuition, the larger the number of samples vs the number of features, the smaller the error P_e

Graphically

For $N_2 \gg N_1$



Let us explain

Something Notable

Let's look at the following example from the paper:

- "A Problem of Dimensionality: A Simple Example" by G.A. Trunk

The Final Goal

The Goal

1. Select the "optimum" number d of features.
2. Select the "best" d features.

The Final Goal

The Goal

1. Select the “optimum” number d of features.
2. Select the “best” d features.

Why? Large d has a three-fold disadvantage

- High computational demands.
- Low generalization performance.
- Poor error estimates

DataLab
Data Science Community

The Final Goal

The Goal

1. Select the "optimum" number d of features.
2. Select the "best" d features.

Why? Large d has a three-fold disadvantage

- High computational demands.
- Low generalization performance.
- Poor error estimates

DataLab
Data Science Community

The Final Goal

The Goal

1. Select the "optimum" number d of features.
2. Select the "best" d features.

Why? Large d has a three-fold disadvantage

- High computational demands.
- Low generalization performance.
- Poor error estimates

DataLab
Data Science Community

The Final Goal

The Goal

1. Select the "optimum" number d of features.
2. Select the "best" d features.

Why? Large d has a three-fold disadvantage

- High computational demands.
- Low generalization performance.
- Poor error estimates

DataLab
Data Science Community

Outline

DataLab
Data Science Community

Back to Feature Selection

Given N

d must be large enough to learn what makes classes different and what makes patterns in the same class similar

In addition

d must be small enough not to learn what makes patterns of the same class different

In practice

In practice, $d < N/3$ has been reported to be a sensible choice for a number of cases

Back to Feature Selection

Given N

d must be large enough to learn what makes classes different and what makes patterns in the same class similar

In addition

d must be small enough not to learn what makes patterns of the same class different

In practice

In practice, $d < N/3$ has been reported to be a sensible choice for a number of cases

DataLab
Data Science Community

Back to Feature Selection

Given N

d must be large enough to learn what makes classes different and what makes patterns in the same class similar

In addition

d must be small enough not to learn what makes patterns of the same class different

In practice

In practice, $d < {}^{N}\!/_{3}$ has been reported to be a sensible choice for a number of cases

Thus

Oh!!!

Once d has been decided, choose the d most informative features:

Best: Large between class distance, Small within class variance.

Thus

Oh!!!

Once d has been decided, choose the d most informative features:

Best: Large between class distance, Small within class variance.

The basic philosophy

1. Discard individual features with poor information content.
2. The remaining information rich features are examined jointly as vectors

DataLab
Data Science Community

Thus

Oh!!!

Once d has been decided, choose the d most informative features:

Best: Large between class distance, Small within class variance.

The basic philosophy

1. Discard individual features with poor information content.
2. The remaining information rich features are examined jointly as vectors

Thus

Oh!!!

Once d has been decided, choose the d most informative features:

Best: Large between class distance, Small within class variance.

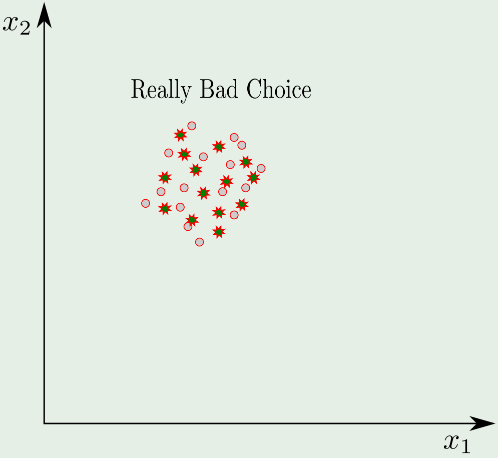
The basic philosophy

1. Discard individual features with poor information content.
2. The remaining information rich features are examined jointly as vectors

DataLab
Data Science Community

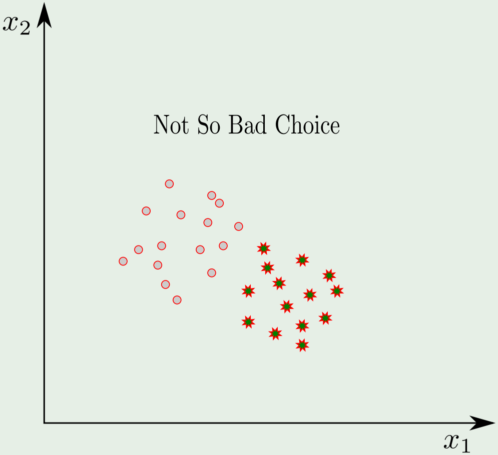
Example

Thus, we want to avoid choices



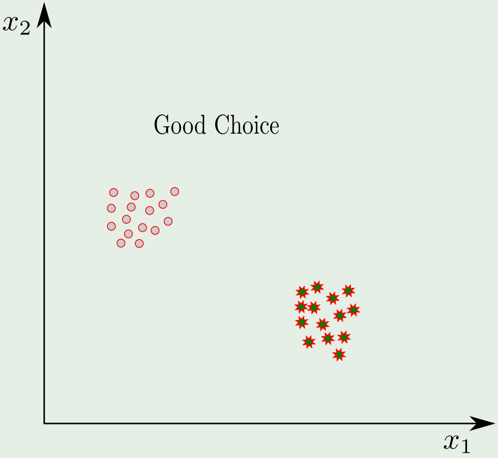
Example

Better Choice



Example

What We Want to Have



Outline

DataLab
Data Science Community

Scatter Matrices

Definition

These are used as a measure of the way data are scattered in the respective feature space.

Scatter Matrices

Definition

These are used as a measure of the way data are scattered in the respective feature space.

Within-class Scatter Matrix

$$S_w = \sum_{i=1}^{C} P_i \Sigma_i \tag{5}$$

where C is the number of classes.

DataLab
Data Science Community

Scatter Matrices

Definition

These are used as a measure of the way data are scattered in the respective feature space.

Within-class Scatter Matrix

$$S_w = \sum_{i=1}^{C} P_i \Sigma_i \tag{5}$$

where C is the number of classes.

Where

We can use the sample mean

$$S_i = E\left[\left(\boldsymbol{x}-\boldsymbol{\mu_i}\right)\left(\boldsymbol{x}-\boldsymbol{\mu_i}\right)^T\right] \approx \frac{1}{N-1}\sum_{k=1}^{n_i}\left(\boldsymbol{x}_{ki}-\boldsymbol{m}_i\right)\left(\boldsymbol{x}_{ki}-\boldsymbol{m}_i\right)^T$$

And P_i the a priori probability of class ω_i defined as

$$P_i \cong n_i/N$$

n_i is the number of samples in class ω_i.

Where

We can use the sample mean

$$S_i = E\left[\left(\boldsymbol{x} - \boldsymbol{\mu_i}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_i}\right)^T\right] \approx \frac{1}{N-1}\sum_{k=1}^{n_i}\left(\boldsymbol{x}_{ki} - \boldsymbol{m}_i\right)\left(\boldsymbol{x}_{ki} - \boldsymbol{m}_i\right)^T$$

And P_i the a priori probability of class ω_i defined as

$$P_i \cong {}^{n_i}/_{N}$$

n_i is the number of samples in class ω_i.

DataLab
Data Science Community

Scatter Matrices

Between-class scatter matrix

$$S_b = \sum_{i=1}^{C} P_i \left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)^T \tag{6}$$

Where

$$\boldsymbol{\mu_0} = \sum_{i=1}^{C} P_i \boldsymbol{\mu}_i \tag{7}$$

The global mean.

Mixture scatter matrix

$$S_m = E\left[\left(\boldsymbol{x} - \boldsymbol{\mu}_0\right)\left(\boldsymbol{x} - \boldsymbol{\mu}_0\right)^T\right] \approx \frac{1}{N-1}\sum_{k=1}^{N}\left(\boldsymbol{x}_i - \boldsymbol{\mu}_0\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_0\right)^T \tag{8}$$

Note: it can be proved that $S_m = S_w + S_b$

Scatter Matrices

Between-class scatter matrix

$$S_b = \sum_{i=1}^{C} P_i \left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)^T \tag{6}$$

Where

$$\boldsymbol{\mu_0} = \sum_{i=1}^{C} P_i \boldsymbol{\mu}_i \tag{7}$$

The global mean.

Mixture scatter matrix

$$S_m = E\left[\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)^T\right] \approx \frac{1}{N-1}\sum_{k=1}^{N}\left(\boldsymbol{x}_i - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu_0}\right)^T \tag{8}$$

Note: it can be proved that $S_m = S_w + S_b$

Scatter Matrices

Between-class scatter matrix

$$S_b = \sum_{i=1}^{C} P_i \left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)^T \tag{6}$$

Where

$$\boldsymbol{\mu_0} = \sum_{i=1}^{C} P_i \boldsymbol{\mu}_i \tag{7}$$

The global mean.

Mixture scatter matrix

$$S_m = E\left[\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)^T\right] \approx \frac{1}{N-1} \sum_{k=1}^{N} \left(\boldsymbol{x}_i - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu_0}\right)^T \tag{8}$$

Note: it can be proved that $S_m = S_w + S_b$

Criterion

The one we can use

$$J_1 = \frac{trace\left\{S_m\right\}}{trace\left\{S_w\right\}} \tag{9}$$

Criterion

The one we can use

$$J_1 = \frac{trace\left\{S_m\right\}}{trace\left\{S_w\right\}} \quad (9)$$

Meaning

It takes takes large values when samples in the d-dimensional space are well clustered around their mean, within each class, and the clusters of the different classes are well separated.

Outline

DataLab
Data Science Community

What to do with it

We want to avoid

High Complexities

What to do with it

We want to avoid

High Complexities

As for example

1. Select a class separability
2. Then, get all possible combinations of features

$$\binom{m}{l}$$

with $l = 1, 2, ..., m$

What to do with it

We want to avoid

High Complexities

As for example

1. Select a class separability
2. Then, get all possible combinations of features

$$\binom{m}{l}$$

with $l = 1, 2, ..., m$

We can do better

1. Sequential Backward Selection

What to do with it

We want to avoid

High Complexities

As for example

1. Select a class separability
2. Then, get all possible combinations of features

$$\binom{m}{l}$$

with $l = 1, 2, ..., m$

We can do better

1. Sequential Backward Selection

DataLab
Data Science Community

What to do with it

We want to avoid

High Complexities

As for example

1. Select a class separability
2. Then, get all possible combinations of features

$$\begin{pmatrix} m \\ l \end{pmatrix}$$

with $l = 1, 2, ..., m$

We can do better

1. Sequential Backward Selection

Outline

For example: Sequential Backward Selection

We have the following example

Given x_1, x_2, x_3, x_4 and we wish to select two of them

Step 1

Adopt a class separability criterion, C, and compute its value for the feature vector $[x_1, x_2, x_3, x_4]^T$.

Step 2

Eliminate one feature, you get

$$[x_1, x_2, x_3]^T, [x_1, x_2, x_4]^T, [x_1, x_3, x_4]^T, [x_2, x_3, x_4]^T,$$

For example: Sequential Backward Selection

We have the following example

Given x_1, x_2, x_3, x_4 and we wish to select two of them

Step 1

Adopt a class separability criterion, C, and compute its value for the feature vector $[x_1, x_2, x_3, x_4]^T$.

Step 2

Eliminate one feature, you get

$$[x_1, x_2, x_3]^T, [x_1, x_2, x_4]^T, [x_1, x_3, x_4]^T, [x_2, x_3, x_4]^T,$$

For example: Sequential Backward Selection

We have the following example

Given x_1, x_2, x_3, x_4 and we wish to select two of them

Step 1

Adopt a class separability criterion, C, and compute its value for the feature vector $[x_1, x_2, x_3, x_4]^T$.

Step 2

Eliminate one feature, you get

$$[x_1, x_2, x_3]^T, [x_1, x_2, x_4]^T, [x_1, x_3, x_4]^T, [x_2, x_3, x_4]^T,$$

For example: Sequential Backward Selection

You use your criterion C

Thus the winner is $[x_1, x_2, x_3]^T$

Step 3

Now, eliminate a feature and generate $[x_1, x_2]^T, [x_1, x_3]^T, [x_2, x_3]^T$.

Use criterion C

To select the best one

For example: Sequential Backward Selection

You use your criterion C

Thus the winner is $[x_1, x_2, x_3]^T$

Step 3

Now, eliminate a feature and generate $[x_1, x_2]^T, [x_1, x_3]^T, [x_2, x_3]^T$,

Use criterion C

To select the best one

For example: Sequential Backward Selection

You use your criterion C

Thus the winner is $[x_1, x_2, x_3]^T$

Step 3

Now, eliminate a feature and generate $[x_1, x_2]^T, [x_1, x_3]^T, [x_2, x_3]^T,$

Use criterion C

To select the best one

Complexity of the Method

Complexity

Thus, starting from m, at each step we drop out one feature from the "best" combination until we obtain a vector of l features.

DataLab
Data Science Community

Complexity of the Method

Complexity

Thus, starting from m, at each step we drop out one feature from the "best" combination until we obtain a vector of l features.

Thus, we need

$1 + 1/2((m+1)m - l(l+1))$ combinations

- It suffers of the so-called nesting-effect
  - Once a feature is discarded, there is no way to reconsider that feature again.

DataLab
Data Science Community

Complexity of the Method

Complexity

Thus, starting from m, at each step we drop out one feature from the "best" combination until we obtain a vector of l features.

Thus, we need

$1 + 1/2((m+1)m - l(l+1))$ combinations

However

- The method is sub-optimal
- It suffers of the so called nesting-effect
  - Once a feature is discarded, there is no way to reconsider that feature again.

Complexity of the Method

Complexity

Thus, starting from m, at each step we drop out one feature from the "best" combination until we obtain a vector of l features.

Thus, we need

$1 + 1/2((m+1)m - l(l+1))$ combinations

However

- The method is sub-optimal
- It suffers of the so called nesting-effect
  - Once a feature is discarded, there is no way to reconsider that feature again.

Complexity of the Method

Complexity

Thus, starting from m, at each step we drop out one feature from the "best" combination until we obtain a vector of l features.

Thus, we need

$1 + 1/2((m+1)m - l(l+1))$ combinations

However

- The method is sub-optimal
- It suffers of the so called nesting-effect
  - Once a feature is discarded, there is no way to reconsider that feature again.

DataLab
Data Science Community