Introduction to Machine Learning

Feature Selection and Generation

Andres Mendez-Vazquez

September 8, 2025

Outline

Outline

Why Feature Engineering?

As always we love simple linear models

- Easy to analyze
- Unique solution

Why Feature Engineering?

As always we love simple linear models

- Easy to analyze
- Unique solution

Definition

- Feature engineering (or feature extraction) is the process of using domain knowledge to extract features (characteristics, properties, attributes) from raw data.

Outline

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.

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.

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

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.

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.

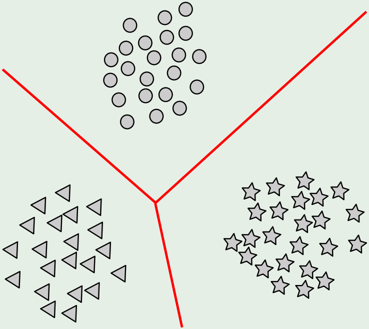
Therefore

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. Outlier removal.

However, Before That...

It is necessary to do the following

1. Outlier removal.
2. Data normalization.

However, Before That...

It is necessary to do the following

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

However, Before That...

It is necessary to do the following

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

Actually

PREPROCESSING!!!

Outline

Outliers

Definition

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

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

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

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

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

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

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

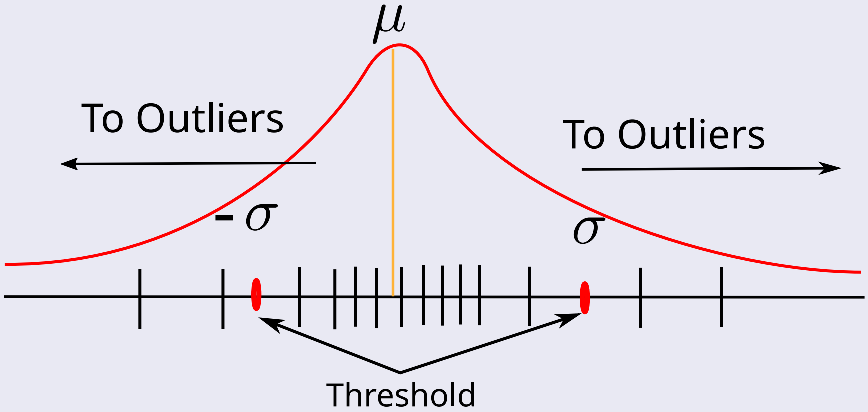
1. A distance of two times the standard deviation covers 95% of the points.
2. A distance of three times the standard deviation covers 99% of the points.

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 measureme

For example, we can use the standard deviation

For a set of samples $x_1, x_2, x_3, \ldots \in \mathbb{R}$



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

Outlier Removal

Important

Then removing outliers is the biggest importance.

Therefore

You can do the following

1. 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

1. If you have a small number $\Rightarrow$ discard them!!!
2. Adopt cost functions that are not sensitive to outliers:

Outlier Removal

Important

Then removing outliers is the biggest importance.

Therefore

You can do the following

1. If you have a small number $\Rightarrow$ discard them!!!
2. Adopt cost functions that are not sensitive to outliers:
3. For more techniques
   1. Huber, P.J. "Robust Statistics," JohnWiley and Sons, 2nd Ed 2009.

Outline

We can do the following

Algorithm

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

Output: Candidate Outliers

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.

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.

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

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

$$\chi_d^2\,(0.05)$$

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.

How?

Get the Sample Mean per feature k

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

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

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

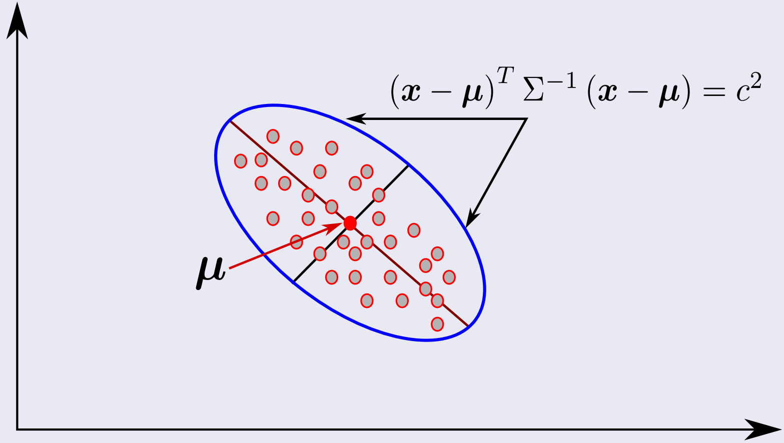
Mahalonobis Distance

We have

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

Thus

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



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)
```

Outline

Data Normalization

In the real world

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

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

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

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.

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.

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.

Outline

Min-Max Method

Be Naive

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

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

Min-Max Method

Be Naive

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

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

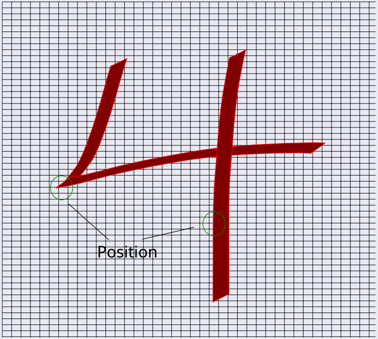
Problem

- This simple normalization will send everything to a unitary sphere!!!
  - However, it works for certain type of data in Deep Learning

However

Even though this can happens there have been report that it can work...

- When data does not depend of single values as:



Gaussian Method

Use the idea of

Everything is Gaussian...

Gaussian Method

Use the idea of

Everything is Gaussian...

Thus

- For each feature set...

Gaussian Method

Use the idea of

Everything is Gaussian...

Thus

- For each feature set...
  1. $\overline{x}_k = \frac{1}{N}\sum_{i=1}^{N} x_{ik},\ k = 1, 2, ..., d$

Gaussian Method

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 Method

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 Mehtod

Thus

- All new features have zero mean and unit variance.

Gaussian Mehtod

Thus

- All new features have zero mean and unit variance.

Further

- Other linear techniques limit the feature values in the range of $[0, 1]$ or $[-1, 1]$ by proper scaling.

Gaussian Mehtod

Thus

- All new features have zero mean and unit variance.

Further

- Other linear techniques limit the feature values in the range of $[0, 1]$ or $[-1, 1]$ by proper scaling.

However

- We can non-linear mapping. For example the softmax scaling.

Soft Max Scaling

Softmax Scaling

- It consists of two steps

Soft Max Scaling

Softmax Scaling

- It consists of two steps

First one

$$y_{ik} = \frac{x_{ik} - \overline{x}_k}{\sigma} \tag{3}$$

Soft Max Scaling

Softmax Scaling

- It consists of two steps

First one

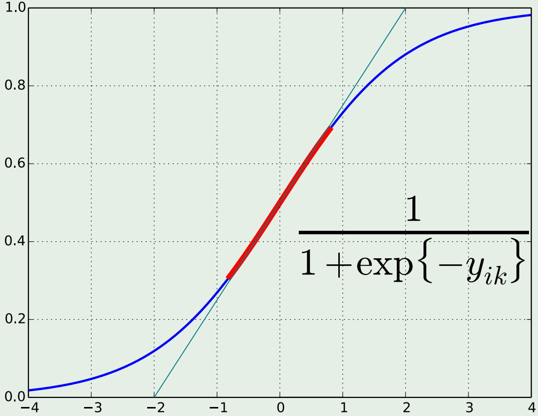
$$y_{ik} = \frac{x_{ik} - \overline{x}_k}{\sigma} \tag{3}$$

Second one

$$\hat{x}_{ik} = \frac{1}{1 + \exp\left\{-y_{ik}\right\}} \tag{4}$$

Explanation

Notice the red area is almost flat!!!



Actually

Thus, we have that

- The red region represents values of y inside of the region defined by the mean and variance (small values of y).
- Then, if we have those values x behaves as a linear function.

Outline

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.

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.

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.

Some traditional techniques to solve this problem

Use zeros and risked it!!!

The idea is not to add anything to the features

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

$$\overline{x}_i = \frac{1}{N} \sum_{k=1}^{N} x_{ik} \tag{5}$$

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

$$\overline{x}_i = \frac{1}{N}\sum_{k=1}^{N} x_{ik} \tag{5}$$

Find the distribution of your data

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

$$\overline{x}_i = \frac{\alpha}{\alpha+\beta} \tag{6}$$

The MOST traditional

Drop it

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

Outline

Example

We have two matrices

- Data Matrix X
- Missing Data M

$$M_{ij} = \begin{cases} 0 & X_{ij} \text{ is missing} \\ 1 & X_{ij} \text{ is not missing} \end{cases}$$

Example

We have two matrices

- Data Matrix X
- Missing Data M

$$M_{ij} = \begin{cases} 0 & X_{ij} \text{ is missing} \\ 1 & X_{ij} \text{ is not missing} \end{cases}$$

Therefore, we have

- $X = (X_{obs}, X_{mis})$

Example

We have two matrices

- Data Matrix X
- Missing Data M

$$M_{ij} = \begin{cases} 0 & X_{ij} \text{ is missing} \\ 1 & X_{ij} \text{ is not missing} \end{cases}$$

Therefore, we have

- $X = (X_{obs}, X_{mis})$

This comes from

- "Bayes and multiple imputation" by RJA Little, DB Rubin (2002)

We can use the following optimization

We can do the following

$$\min_{M_{ij}=1} \|X - AB\|_F$$

We can use the following optimization

We can do the following

$$\min_{M_{ij}=1} \|X - AB\|_F$$

Clearly an initial matrix decomposition, where

$$M_{ij}x_{ij} \approx \sum_{k=1}^{K} a_{ik}b_{kj}$$

We can use the following optimization

We can do the following

$$\min_{M_{ij}=1} \|X - AB\|_F$$

Clearly an initial matrix decomposition, where

$$M_{ij}x_{ij} \approx \sum_{k=1}^{K} a_{ik}b_{kj}$$

So the total error to be minimized is

$$\min_{M_{ij}=1} \|X - AB\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M}\left[M_{ij}x_{ij} - \sum_{k=1}^{K} a_{ik}b_{kj}\right]^2}$$

- $K \ll N, M$

This can be regularized

Using the following ideas

$$\min_{M_{ij}=1} \|X - AB\|_F + \lambda \left[\|A\|^2 + \|B\|^2 \right]$$

This can be regularized

Using the following ideas

$$\min_{M_{ij}=1} \|X - AB\|_F + \lambda \left[\|A\|^2 + \|B\|^2 \right]$$

Therefore, once the minimization is achieved

- We finish with two dense matrices A, B that can be used to obtain the elements with entries $M_{ij} = 0$

Outline

THE PEAKING PHENOMENON

Remeber

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.

THE PEAKING PHENOMENON

Remeber

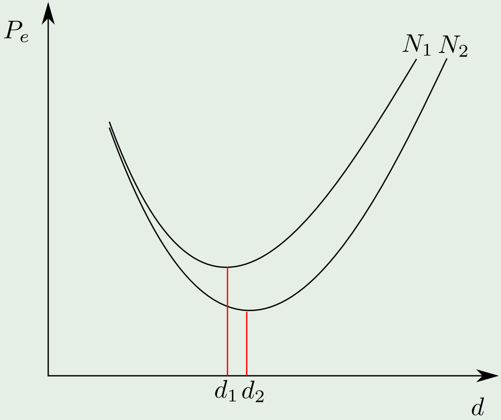
Normally, to design a classifier with good generalization performance, we want the number of sample N to be larger than the number of features d.

What?

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

Graphically

For $N_2 \gg N_1$



The Goal of Feature Selection

The Goal

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

The Goal of Feature Selection

The Goal

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

The Goal of Feature Selection

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.

The Goal of Feature Selection

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.

The Goal of Feature Selection

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

Outline

Feature Selection

Given N

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

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

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:

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.

Thus

Oh!!!

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

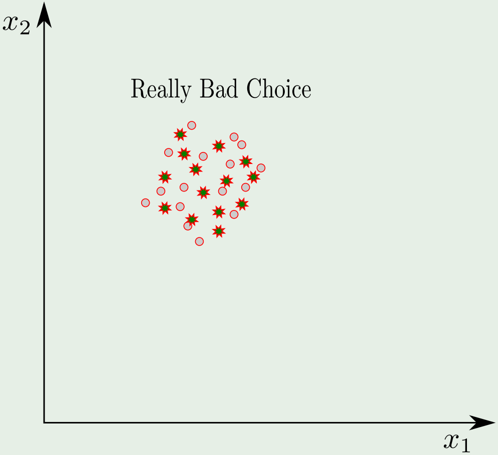
Best: Large between class distance, Small within class variance.

The basic philosophy

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

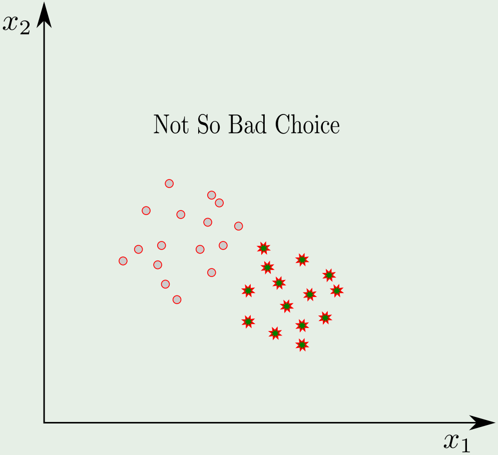
Example

Thus, we want to avoid choices



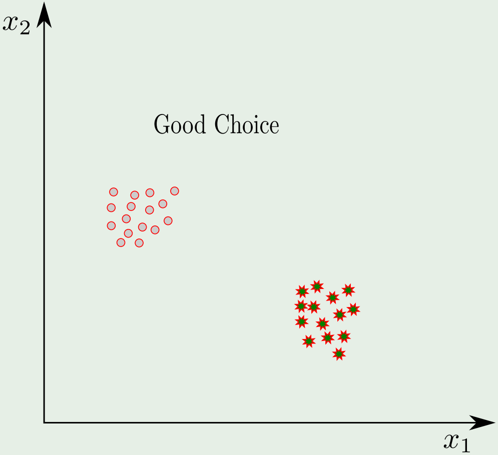
Example

Better Choice



Example

What We Want to Have



Outline

Considering Feature Sets

Something Notable

- The emphasis so far was on individually considered features.

Considering Feature Sets

Something Notable

- The emphasis so far was on individually considered features.

But

- That is, two features may be rich in information, but if they are highly correlated we need not consider both of them.

Considering Feature Sets

Something Notable

- The emphasis so far was on individually considered features.

But

- That is, two features may be rich in information, but if they are highly correlated we need not consider both of them.

Then

- Combine features to search for the "best" combination after features have been discarded.

What to do?

Possible

- Use different feature combinations to form the feature vector.

What to do?

Possible

- Use different feature combinations to form the feature vector.
- Train the classifier, and choose the combination resulting in the best classifier performance.

What to do?

Possible

- Use different feature combinations to form the feature vector.
- Train the classifier, and choose the combination resulting in the best classifier performance.

However

- A major disadvantage of this approach is the high complexity.

What to do?

Possible

- Use different feature combinations to form the feature vector.
- Train the classifier, and choose the combination resulting in the best classifier performance.

However

- A major disadvantage of this approach is the high complexity.
- Also, local minimum may give misleading results.

What to do?

Possible

- Use different feature combinations to form the feature vector.
- Train the classifier, and choose the combination resulting in the best classifier performance.

However

- A major disadvantage of this approach is the high complexity.
- Also, local minimum may give misleading results.

Better

- Adopt a class separability measure and choose the best feature combination against this cost.

Outline

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

$$S_w = \sum_{i=1}^{C} P_i S_i \tag{7}$$

- where C is the number of classes.

Scatter Matrices

Definition

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

Within-class Scatter Matrix

$$S_w = \sum_{i=1}^{C} P_i S_i \tag{7}$$

- where C is the number of classes.

where

1. $S_i = E\left[(\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^T\right]$

Scatter Matrices

Definition

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

Within-class Scatter Matrix

$$S_w = \sum_{i=1}^{C} P_i S_i \tag{7}$$

- where C is the number of classes.

where

1. $S_i = E\left[(\boldsymbol{x} - \boldsymbol{\mu_i})(\boldsymbol{x} - \boldsymbol{\mu_i})^T\right]$
2. P_i the a priori probability of class ω_i defined as $P_i \cong {}^{n_i}/_{N}$.

Scatter Matrices

Definition

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

Within-class Scatter Matrix

$$S_w = \sum_{i=1}^{C} P_i S_i \tag{7}$$

- where C is the number of classes.

where

1. $S_i = E\left[\left(\boldsymbol{x} - \boldsymbol{\mu_i} \right) \left(\boldsymbol{x} - \boldsymbol{\mu_i} \right)^T \right]$
2. P_i the a priori probability of class ω_i defined as $P_i \cong n_i / N$.
   1. n_i is the number of samples in class ω_i.

Scatter Matrices

Between-class scatter matrix

$$S_b = \sum_{i=1}^{C} P_i \left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)^T \tag{8}$$

Scatter Matrices

Between-class scatter matrix

$$S_b = \sum_{i=1}^{C} P_i \left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)^T \tag{8}$$

Where

$$\boldsymbol{\mu_0} = \sum_{i=1}^{C} P_i \boldsymbol{\mu}_i \tag{9}$$

The global mean.

Scatter Matrices

Between-class scatter matrix

$$S_b = \sum_{i=1}^{C} P_i \left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)^T \tag{8}$$

Where

$$\boldsymbol{\mu_0} = \sum_{i=1}^{C} P_i \boldsymbol{\mu}_i \tag{9}$$

The global mean.

Mixture scatter matrix

$$S_m = E\left[\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)\left(\boldsymbol{x} - \boldsymbol{\mu_0}\right)^T\right] \tag{10}$$

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

Criterion's

First One

$$J_1 = \frac{trace\left\{S_m\right\}}{trace\left\{S_w\right\}} \quad (11)$$

- 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.

Criterion's

First One

$$J_1 = \frac{trace\left\{S_m\right\}}{trace\left\{S_w\right\}} \tag{11}$$

- 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.

Other Criteria are

1. $J_2 = \frac{|S_m|}{|S_w|}$

Criterion's

First One

$$J_1 = \frac{trace\left\{S_m\right\}}{trace\left\{S_w\right\}} \tag{11}$$

- 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.

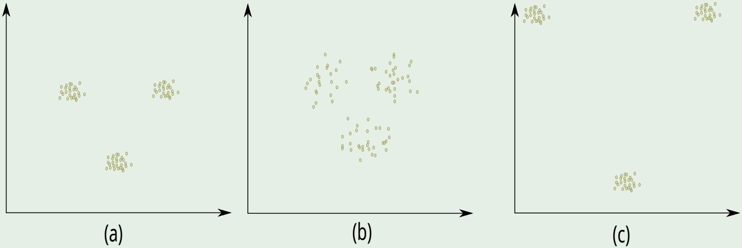
Other Criteria are

1. $J_2 = \frac{|S_m|}{|S_w|}$
2. $J_3 = trace\left\{S_w^{-1} S_m\right\}$

Example

We have

- Classes with
  - (a) small within-class variance and small between-class distances,
  - (b) large within- class variance and small between-class distances,
  - (c) small within-class variance and large between-class distances.



Outline

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

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

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

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
2. Sequential Forward 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

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

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

We can do better

1. Sequential Backward Selection
2. Sequential Forward Selection
3. Floating Search Methods

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
2. Sequential Forward Selection
3. Floating Search Methods

However these are sub-optimal methods

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

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

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 $[x_1, x_2, x_3]^T$

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

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.

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

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

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

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.

Similar Problem

For

- Sequential Forward Selection

Similar Problem

For

- Sequential Forward Selection

We can overcome this by using

- Floating Search Methods

Similar Problem

For

- Sequential Forward Selection

We can overcome this by using

- Floating Search Methods

A more elegant methods are the ones based on

- Dynamic Programming
- Branch and Bound

Outline

What do we want?

What

- Given a set of measurements, the goal is to discover compact and informative representations of the obtained data.

What do we want?

What

- 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, leading to a reduction of the necessary feature space dimension.

What do we want?

What

- 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, leading to a reduction of the necessary feature space dimension.

Properties

- Thus removing information redundancies - Usually produced and the measurement.

What Methods we will see?

Fisher Linear Discriminant

1. Squeezing to the maximum.
2. From Many to One Dimension

What Methods we will see?

Fisher Linear Discriminant

1. Squeezing to the maximum.
2. From Many to One Dimension

Principal Component Analysis

1. Not so much squeezing
2. You are willing to lose some information

Outline

Did you noticed?

That Rotations really do not exist

- Actually, they are mappings or projections in linear algebra

Did you noticed?

That Rotations really do not exist

- Actually, they are mappings or projections in linear algebra

Thus, Can we get more powerful mappings?

- To obtain better features

Did you noticed?

That Rotations really do not exist

- Actually, they are mappings or projections in linear algebra

Thus, Can we get more powerful mappings?

- To obtain better features

Clearly... Yes

- For example, Principal Components

Also Known as Karhunen-Loeve Transform

Setup

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

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)

Basically

Principal Component Analysis

- Attempts to maximize the variance in certain vectors

Basically

Principal Component Analysis

- Attempts to maximize the variance in certain vectors

Basically Linear Algebra

- Basically discover the basis that describe best the data dispersion in specific directions

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{12}$$

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

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{12}$$

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

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i \tag{13}$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{12}$$

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

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{13}$$

Center data

$$\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}} \tag{14}$$

Build the Sample Mean

The Covariance Matrix

$$S=\frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)^T \tag{15}$$

Build the Sample Mean

The Covariance Matrix

$$S=\frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)^T \tag{15}$$

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.

Outline

Clearly

We need to build a projection

- Remember a square matrix is basically a projection

$$A\boldsymbol{x} = \boldsymbol{x}' \left\{ \text{Projections into the Column Space} \right.$$

Clearly

We need to build a projection

- Remember a square matrix is basically a projection

$$A\boldsymbol{x} = \boldsymbol{x}' \left\{ \text{Projections into the Column Space} \right.$$

Thus, we want to have the larger dispersion's

- Why not start with a column space of a single dimension $==$ single vector

Outline

We have the following steps

Determine covariance matrix

$$S = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T \tag{16}$$

We have the following steps

Determine covariance matrix

$$S = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T \tag{16}$$

Generate the decomposition

$$S = U\Sigma U^T$$

We have the following steps

Determine covariance matrix

$$S=\frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)^T \tag{16}$$

Generate the decomposition

$$S=U\Sigma U^T$$

With

- Eigenvalues in Σ and eigenvectors in the columns of U.

Then

Project samples $\boldsymbol{x}_i$ into subspaces dim$=k$

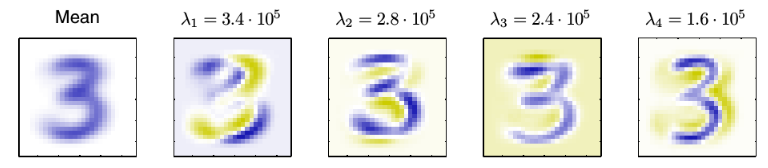
$$z_i = U_K^T \boldsymbol{x}_i$$

- With U_k is a matrix with k columns

Outline

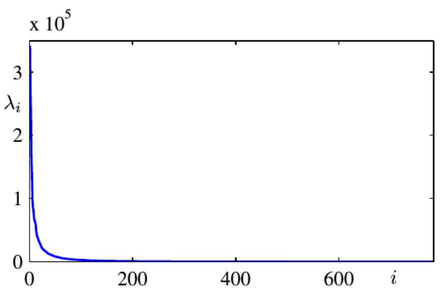
Example

From Bishop



Example

From Bishop



Example

From Bishop