Aalto University
School of Electrical Engineering

CS-C3240 – Machine Learning D

Feature Engineering

Dariush Salami

Department of Communications and Networking
Aalto University, School of Electrical Engineering
dariush.salami@aalto.fi

Version 1.0, September 15, 2022

Learning goals

Understand the concepts of

- feature engineering
- feature selection
- challenges with high dimensional feature spaces
- Principle Component Analysis
- Kernel methods

Outline

Feature Engineering

Strategies to cope with common challenges

Principle Component Analysis

Kernel methods

Feature engineering

Feature pre-processing

→ Normalisation

→ Detection of outliers

→ Are features independent?

Feature engineering

Simple normalization: Scaling

For each sample x_i from a set $\mathcal{X}$, compute the scaled value as

$$x_i' = \frac{x_i - \min(\mathcal{X})}{\max(\mathcal{X}) - \min(\mathcal{X})}$$

Feature pre-processing

→ Normalisation
→ Detection of outliers
→ Are features independent?

Feature engineering

Simple normalization: Scaling

For each sample x_i from a set $\mathcal{X}$, compute the scaled value as

$$x_i' = \frac{x_i - \min(\mathcal{X})}{\max(\mathcal{X}) - \min(\mathcal{X})}$$

after scaling, it is common to center the values around e.g. 0 or their arithmetic mean, median, centre of mass etc.

Feature pre-processing

→ Normalisation

→ Detection of outliers

→ Are features independent?

Feature engineering

Standardization to zero mean/unit variance

Given a set of values $x_i; i \in \{1..n\}$ from a set $\mathcal{X}$ with mean μ and standard deviation σ, we derive the standardized values x_i' as

$$x_i' = \frac{x_i - \mu}{\sigma}$$

Feature pre-processing

$\rightarrow$ Normalisation

$\rightarrow$ Detection of outliers

$\rightarrow$ Are features independent?

Feature engineering

Standardization to zero mean/unit variance

Given a set of values $x_i; i \in \{1..n\}$ from a set $\mathcal{X}$ with mean μ and standard deviation σ, we derive the standardized values x_i' as

$$x_i' = \frac{x_i - \mu}{\sigma}$$

Using the variance σ^2 instead of σ is called variance scaling

Feature pre-processing

→ Normalisation

→ Detection of outliers

→ Are features independent?

Feature engineering

Important:

When normalizing on the training set input, this need to be applied identically ot the test set input. Do not normalize the test set input on the test set data.

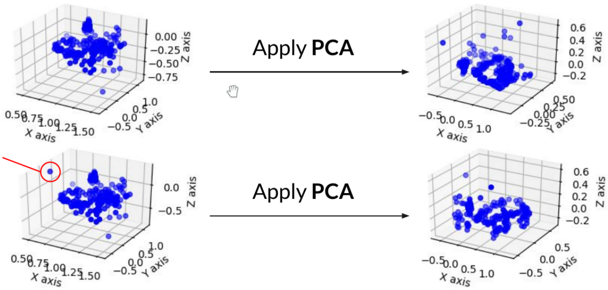
Feature pre-processing

→ Normalisation

→ Detection of outliers

→ Are features independent?

Feature engineering



Feature pre-processing

→ Normalisation

→ Detection of outliers

→ Are features independent?

Feature engineering

Common pitfalls in outlier handling:

It is not unusual to find values that clearly depart from the rest.

Example: In insurance, most claims are small but a few are large. Removing the large claims will completely invalidate an insurance model.

Feature pre-processing

→ Normalisation

→ Detection of outliers

→ Are features independent?

Feature engineering

Common pitfalls in outlier handling:

It is not unusual to find values that clearly depart from the rest.

Example: In insurance, most claims are small but a few are large. Removing the large claims will completely invalidate an insurance model.

Caution: Do <u>not</u> throw away outliers, unless you have evidence that they are errors

Darell Huff, How to lie with Statistics, 1954

Feature pre-processing

→ Normalisation

→ Detection of outliers

→ Are features independent?

Feature engineering

Common pitfalls in outlier handling:

It is not unusual to find values that clearly depart from the rest.

Approach: If outliers are present, use algorithms that are robust to outliers. For instance, covarianceor mean are sensitive to outliers. $\rightarrow$ replace mean with median.

Feature pre-processing

$\rightarrow$ Normalisation

$\rightarrow$ Detection of outliers

$\rightarrow$ Are features independent?

Feature engineering

Common pitfalls in outlier handling:

It is not unusual to find values that clearly depart from the rest.

$\rightarrow$ Outliers behave sometimes different than the rest $\rightarrow$ train separate model on outliers

Detection clustering, density estimation,

Feature pre-processing

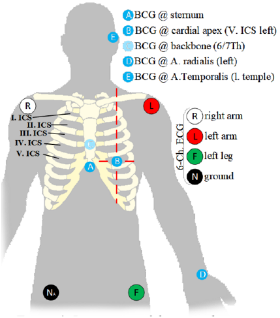
$\rightarrow$ Normalisation

$\rightarrow$ Detection of outliers

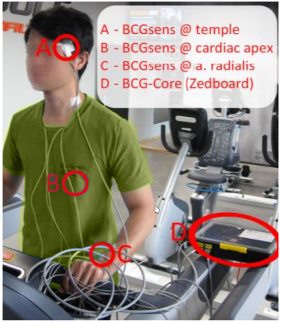
$\rightarrow$ Are features independent?

Feature engineering

Example: walking speed vs. heart rate



(a) Positioning of the sensors



(b) Subject performing the study

Feature pre-processing

→ Normalisation

→ Detection of outliers

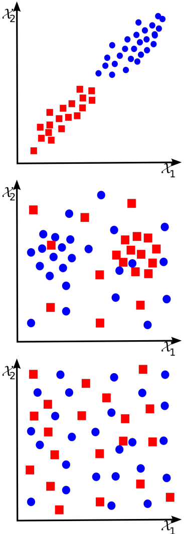
→ Are features independent?

Feature Selection

A large portion of the performance of Machine Learning algorithms is due to the right choice and processing of features.

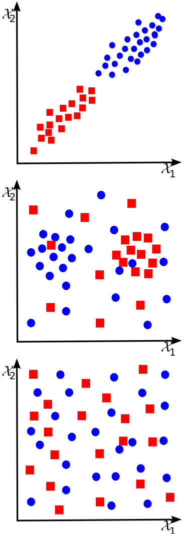
Avoid non-important features

- Noisy data
- Non-correlation between features and classes
- Correlated features
- Sometimes, less is better



Feature Selection

A large portion of the performance of Machine Learning algorithms is due to the right choice and processing of features.



Avoid non-important features

- Noisy data
- Non-correlation between features and classes
- Correlated features
- Sometimes, less is better

Choosing the most important features

- Reduces training and evaluation time
- Reduces complexity of a model (easier to interpret)
- Improves prediction/recall of a model
- Reduces overfitting

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

Las Vegas Filter

Repeatedly generate random feature subsets $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$, train a classifier $\hat{h}_s(\overrightarrow{\hat{w}_s}, \cdot) = \min_{i\in\{\mathcal{X}_s\}} \mathcal{L}\left(h(\overrightarrow{w}, \overrightarrow{x}^{(i)}), y^{(i)}\right)$ and validate $\hat{h}_s(\overrightarrow{\hat{w}_s}, \cdot)$ for its classification performance

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

Focus algorithm

1. Train and evaluate a classifier for singleton feature $\mathcal{X}_o$
2. Evaluate each set of two features $\mathcal{X}_o, \mathcal{X}_p$

$\vdots$

Until consistent solution is found

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

Focus algorithm

1. Train and evaluate a classifier for singleton feature $\mathcal{X}_o$
2. Evaluate each set of two features $\mathcal{X}_o, \mathcal{X}_p$

$\vdots$

Until consistent solution is found

Complexity:

$$\binom{|\mathcal{X}|}{k} = \frac{|\mathcal{X}|!}{(|\mathcal{X}|-k)!(k!)} \rightarrow \mathcal{O}(2^{|\mathcal{X}|})$$

$$\binom{|\mathcal{X}|}{1} \cdot \binom{|\mathcal{X}|}{2} \cdots \binom{|\mathcal{X}|}{|\mathcal{X}|}$$

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

Relief algorithm

Given a collection of values $x_i; i \in \{1..n\}$ of a feature $\mathcal{X}$, compute

$$\frac{\text{Closest distance to all other samples of the same class}}{\text{Closest distance to all samples not in that class}}$$

Rationale: Feature more relevant the more it separates a sample from samples in other classes and the less it separates from samples in same class

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

Relief algorithm

Given a collection of values $x_i; i \in \{1..n\}$ of a feature $\mathcal{X}$, compute

$$\frac{\text{Closest distance to all other samples of the same class}}{\text{Closest distance to all samples not in that class}}$$

Complexity:
$\mathcal{O}\left(|\mathcal{X}| \cdot n^2\right)$

Rationale: Feature more relevant the more it separates a sample from samples in other classes and the less it separates from samples in same class

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

Pearson Correlation Coefficient

$$r(\mathcal{X}_1, \mathcal{X}_2) = \frac{\mathrm{Cov}(\mathcal{X}_1, \mathcal{X}_2)}{\sqrt{\mathrm{Var}(\mathcal{X}_1)\mathrm{Var}(\mathcal{X}_2)}}$$

- Identifies linear relation between features $\mathcal{X}_i$

Feature selection algorithms

How to identify good/meaningful features?

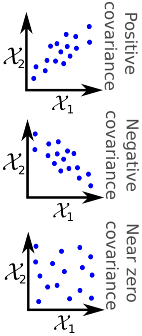
Feature selection

For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

Pearson Correlation Coefficient

$$r(\mathcal{X}_1, \mathcal{X}_2) = \frac{\mathrm{Cov}(\mathcal{X}_1, \mathcal{X}_2)}{\sqrt{\mathrm{Var}(\mathcal{X}_1)\mathrm{Var}(\mathcal{X}_2)}}$$

- Identifies linear relation between features $\mathcal{X}_i$



Feature selection algorithms

How to identify good/meaningful features?

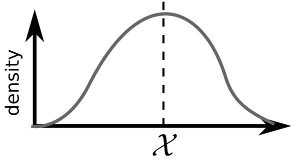
Feature selection

For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

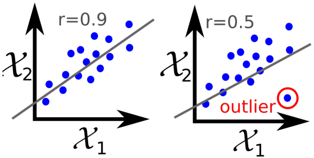
Pearson Correlation Coefficient

$$r(\mathcal{X}_1, \mathcal{X}_2) = \frac{\mathrm{Cov}(\mathcal{X}_1, \mathcal{X}_2)}{\sqrt{\mathrm{Var}(\mathcal{X}_1)\mathrm{Var}(\mathcal{X}_2)}}$$

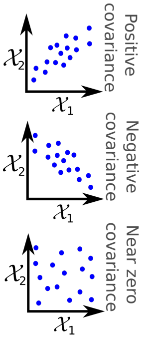
- Identifies linear relation between features $\mathcal{X}_i$



All features should follow a normal distribution



Data should have no significant outliers



Feature selection algorithms

How to identify good/meaningful features?

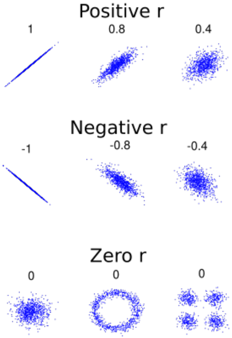
Feature selection

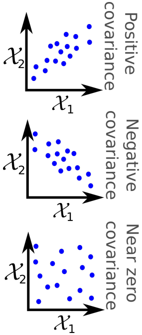
For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{Y}_i \in \{\mathcal{Y}\}$?

Pearson Correlation Coefficient

$$r(\mathcal{X}_1, \mathcal{X}_2) = \frac{\mathrm{Cov}(\mathcal{X}_1, \mathcal{X}_2)}{\sqrt{\mathrm{Var}(\mathcal{X}_1)\mathrm{Var}(\mathcal{X}_2)}}$$

- Identifies linear relation between features $\mathcal{X}_i$





Outline

Feature Engineering

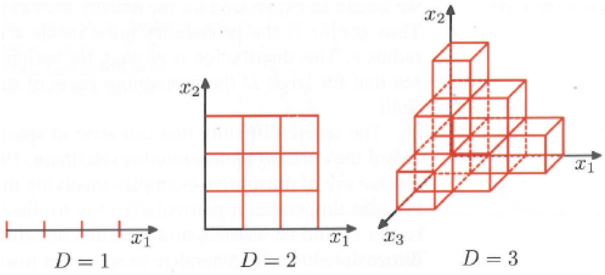
Strategies to cope with common challenges

Principle Component Analysis

Kernel methods

Issues related to high dimensional input data

Exponential growth Volume of the space grows exponentially with dimension

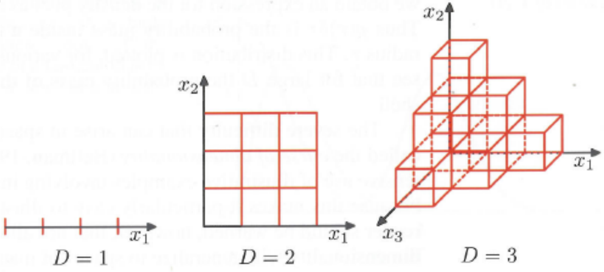


Issues related to high dimensional input data

Exponential growth Volume of the space grows exponentially with dimension

Curse of dimensionality

Too sparse samples across regions to estimate a distribution in that space
(Problematic for methods that require statistical significance)



Issues related to high dimensional input data

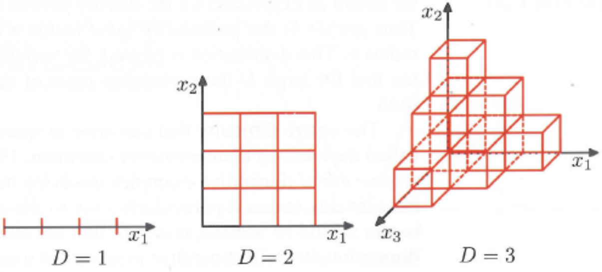
Exponential growth Volume of the space grows exponentially with dimension

Curse of dimensionality

Too sparse samples across regions to estimate a distribution in that space
(Problematic for methods that require statistical significance)

Hughes (peaking) phenomenon

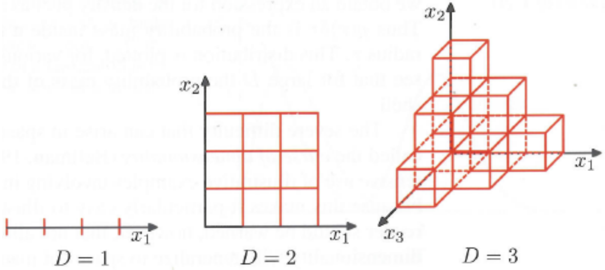
Predictive power of classifier first increases with dimension, then decreases



Issues related to high dimensional input data

Exponential growth Volume of the space grows exponentially with dimension

Counter-intuitive properties in higher dimensional spaces



Issues related to high dimensional input data

Exponential growth Volume of the space grows exponentially with dimension

Counter-intuitive properties in higher dimensional spaces

Example – Volume of a sphere

Consider a sphere of radius $r = 1$ in a D-dimensional space

Issues related to high dimensional input data

Exponential growth Volume of the space grows exponentially with dimension
Counter-intuitive properties in higher dimensional spaces

Example – Volume of a sphere

Consider a sphere of radius $r = 1$ in a D-dimensional space

Fraction of the volume between radius $r = 1$ and $r' = 1 - \varepsilon$?

Issues related to high dimensional input data

Exponential growth Volume of the space grows exponentially with dimension

Counter-intuitive properties in higher dimensional spaces

Example – Volume of a sphere

Consider a sphere of radius $r = 1$ in a D-dimensional space

Fraction of the volume between radius $r = 1$ and $r' = 1 - \varepsilon$?

Volume of shpere with radius r:

$V_D(r) = \delta_D r^D$ for appropriate δ_D

Issues related to high dimensional input data

Exponential growth Volume of the space grows exponentially with dimension

Counter-intuitive properties in higher dimensional spaces

Example – Volume of a sphere

Consider a sphere of radius $r = 1$ in a D-dimensional space

Fraction of the volume between radius $r = 1$ and $r' = 1 - \varepsilon$?

Volume of shpere with radius r:

$$V_D(r) = \delta_D r^D$$

for appropriate δ_D

Given by

$$\frac{V_D(1) - V_D(1-\varepsilon)}{V_D(1)} = 1 - (1-\varepsilon)^D$$

Issues related to high dimensional input data

Exponential growth Volume of the space grows exponentially with dimension
Counter-intuitive properties in higher dimensional spaces

Example – Volume of a sphere

Consider a sphere of radius $r = 1$ in a D-dimensional space

Fraction of the volume between radius $r = 1$ and $r' = 1 - \varepsilon$?

Volume of shpere with radius r:

$$V_D(r) = \delta_D r^D$$

for appropriate δ_D

Given by

$$\frac{V_D(1) - V_D(1-\varepsilon)}{V_D(1)} = 1 - (1-\varepsilon)^D$$

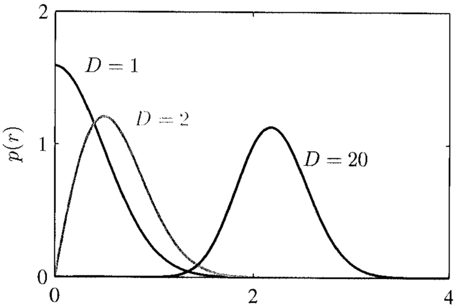
For large D, this fraction tends to 1

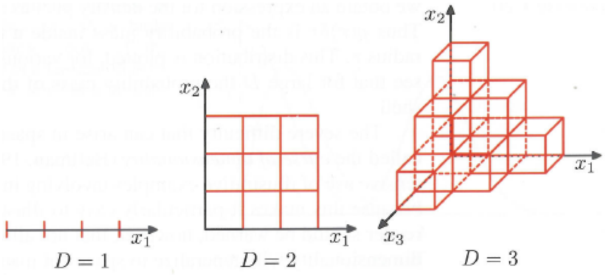
In high dimensions, most of the volume of a sphere concentrates near the surface

Issues related to high dimensional input data

Example – Gaussian distribution

Probability mass concentrated in a thin shell
(here plotted as distance from the origin in a polar coordinate system)

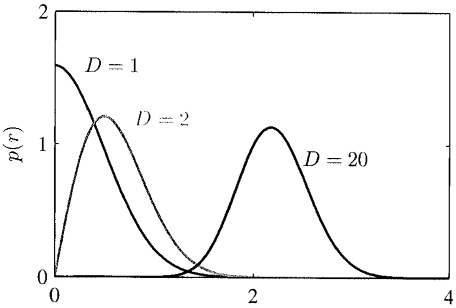




Issues related to high dimensional input data

Example – Gaussian distribution

Probability mass concentrated in a thin shell
(here plotted as distance from the origin in a polar coordinate system)



Curse of Dimensionality

Mechanisms to efficiently reduce dimensions or classifiers that respect properties of high-dimensional spaces required.

Outline

Feature Engineering

Strategies to cope with common challenges

Principle Component Analysis

Kernel methods

Principle Component Analysis

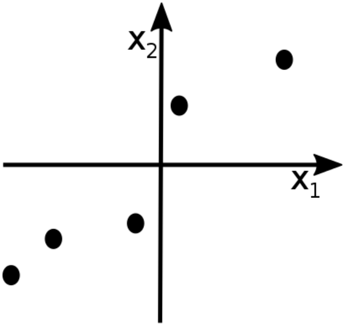
Principal Component Analysis

Find lower dimensional surface onto which to project the data

Principle Component Analysis

Principal Component Analysis

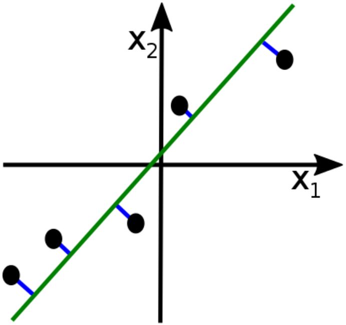
Find lower dimensional surface onto which to project the data



Principle Component Analysis

Principal Component Analysis

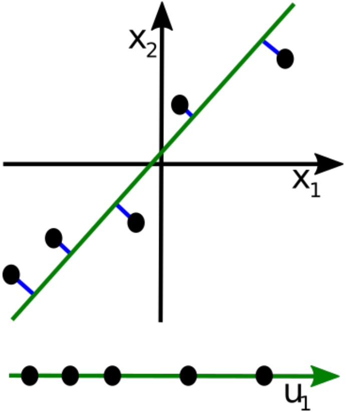
Find lower dimensional surface onto which to project the data



Principle Component Analysis

Principal Component Analysis

Find lower dimensional surface onto which to project the data



Principle Component Analysis

PCA finds k vectors $\overrightarrow{u_1}, \ldots, \overrightarrow{u_k}$ onto which to project the data such that the projection error is minimized.

Principle Component Analysis

PCA finds k vectors $\overrightarrow{u_1}, \ldots, \overrightarrow{u_k}$ onto which to project the data such that the projection error is minimized.

$\rightarrow$ In particular, find $\overrightarrow{z_i} = z_i^{(1)} \ldots z_i^{(n)}$ to represent the $\overrightarrow{x_i} = x_i^{(1)} \ldots x_i^{(n)}$ in this k-dimensional vector space spanned by the $\overrightarrow{u_i}$

Principle Component Analysis

1. Compute the covariance matrix from the $x^{(i)}$:

$$C = \frac{1}{n} \underbrace{\underbrace{\boldsymbol{X}}_{\boldsymbol{n} \times \boldsymbol{m}\text{-dim.}} \underbrace{\boldsymbol{X}^{\boldsymbol{T}}}_{\boldsymbol{m} \times \boldsymbol{n}\text{-dim.}}}_{m \times m\text{-dim.}}$$

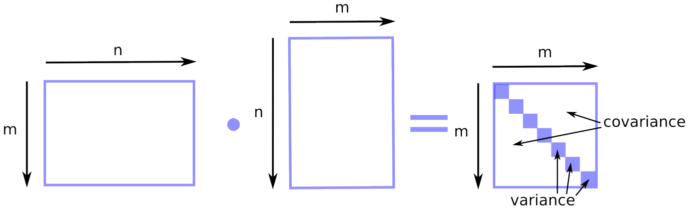
(We assume that all features are mean-normalized and scaled into $[0, 1]$)

Principle Component Analysis

1. Compute the covariance matrix from the $x^{(i)}$:

$$C = \frac{1}{n} \underbrace{\underbrace{\boldsymbol{X}}_{\boldsymbol{n} \times \boldsymbol{m}\text{-dim.}} \underbrace{\boldsymbol{X}^T}_{\boldsymbol{m} \times \boldsymbol{n}\text{-dim.}}}_{m \times m\text{-dim.}}$$

(We assume that all features are mean-normalized and scaled into [0, 1])



Principle Component Analysis

1. Compute the covariance matrix from the $x^{(i)}$:

$$C = \frac{1}{n} \underbrace{\underbrace{\boldsymbol{X}}_{\boldsymbol{n} \times \boldsymbol{m}\text{-dim.}} \underbrace{\boldsymbol{X}^T}_{\boldsymbol{m} \times \boldsymbol{n}\text{-dim.}}}_{m \times m\text{-dim.}}$$

(We assume that all features are mean-normalized and scaled into [0, 1])

Covariance

A measure of spread of a set of points around their center of mass

Principle Component Analysis

1. Compute the covariance matrix from the $x^{(i)}$:

$$C = \frac{1}{n} \underbrace{\underbrace{\boldsymbol{X}}_{\boldsymbol{n}\times\boldsymbol{m}\text{-dim.}} \underbrace{\boldsymbol{X}^{\boldsymbol{T}}}_{\boldsymbol{m}\times\boldsymbol{n}\text{-dim.}}}_{m\times m\text{-dim.}}$$

   (We assume that all features are mean-normalized and scaled into $[0, 1]$)

2. The pricipal components are found by computing the eigenvectors and eigenvalues of C (solving $(C - \lambda I_m)u = 0$)

Principle Component Analysis

When a matrix C is multiplied with a vector u', this usually results in a new vector Cu' of different direction than u'.

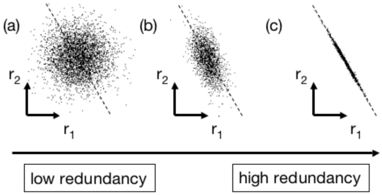
Principle Component Analysis

When a matrix C is multiplied with a vector u', this usually results in a new vector Cu' of different direction than u'.

→ There are few vectors u, however, which have the same direction ($Cu = \lambda u$).

These are the eigenvectors of C and λ are the eigenvalues

Principle Component Analysis



1. Compute the covariance matrix from the $x^{(i)}$:

$$C = \frac{1}{n} \underbrace{\underbrace{\mathbf{X}}_{\mathbf{n}\times\mathbf{m}\text{-dim.}} \underbrace{\mathbf{X}^T}_{\mathbf{m}\times\mathbf{n}\text{-dim.}}}_{m\times m\text{-dim.}}$$

(We assume that all features are mean-normalized and scaled into $[0, 1]$)

2. The pricipal components are found by computing the eigenvectors and eigenvalues of C (solving $(C - \lambda I_m)u = 0$)

Eigenvectors and Eigenvalues

The (orthogonal) eigenvectors are sorted by their eigenvalues with respect to the direction of greatest variance in the data.

Principle Component Analysis

1. Compute the covariance matrix from the $x^{(i)}$:

$$C = \frac{1}{n} \underbrace{\underbrace{\boldsymbol{X}}_{\boldsymbol{n}\times\boldsymbol{m}\text{-dim.}} \underbrace{\boldsymbol{X}^{\boldsymbol{T}}}_{\boldsymbol{m}\times\boldsymbol{n}\text{-dim.}}}_{m\times m\text{-dim.}}$$

   (We assume that all features are mean-normalized and scaled into $[0, 1]$)

2. The pricipal components are found by computing the eigenvectors and eigenvalues of C (solving $(C - \lambda I_m)u = 0$)
3. Choose the k eigenvectors with largest eigenvalues to represent the projection space U

Principle Component Analysis

1. Compute the covariance matrix from the $x^{(i)}$:

$$C = \frac{1}{n} \underbrace{\underbrace{\boldsymbol{X}}_{\boldsymbol{n}\times\boldsymbol{m}\text{-dim.}} \underbrace{\boldsymbol{X}^{\boldsymbol{T}}}_{\boldsymbol{m}\times\boldsymbol{n}\text{-dim.}}}_{m\times m\text{-dim.}}$$

   (We assume that all features are mean-normalized and scaled into $[0, 1]$)

2. The pricipal components are found by computing the eigenvectors and eigenvalues of C (solving $(C - \lambda I_m)u = 0$)
3. Choose the k eigenvectors with largest eigenvalues to represent the projection space U
4. These k eigenvectors in U are used to transform the inputs x_i to z_i:

$$z^{(i)} = U^T x^{(i)}$$

Principle Component Analysis

How to choose the number k of dimensions?

We can calculate

$$\frac{\text{Average squared projection error}}{\text{Total variation in the data}} \rightarrow \frac{\sum_{i=1}^{k} ||x^{(i)} - x_{\text{approx}}^{(i)}||^2}{\frac{1}{m}\sum_{i=1}^{m} ||x^{(i)}||^2}$$

as the accuracy of the projection using k principle components as a function of the eigenvalues

$$\frac{\sum_{i=1}^{k} \sqrt{\lambda_i}}{\sum_{j=1}^{m} \sqrt{\lambda_j}} = d$$

Principle Component Analysis

How to choose the number *k* of dimensions?

We can calculate

$$\frac{\text{Average squared projection error}}{\text{Total variation in the data}} \rightarrow \frac{\sum_{i=1}^{k} ||x^{(i)} - x_{\text{approx}}^{(i)}||^2}{\frac{1}{m}\sum_{i=1}^{m} ||x^{(i)}||^2}$$

as the accuracy of the projection using *k* principle components as a function of the eigenvalues

$$\frac{\sum_{i=1}^{k} \sqrt{\lambda_i}}{\sum_{j=1}^{m} \sqrt{\lambda_j}} = d$$

We say that $100 \cdot (1 - d)\%$ of variance is retained.

(Typically, $d \in [0.01, 0.05]$)

Outline

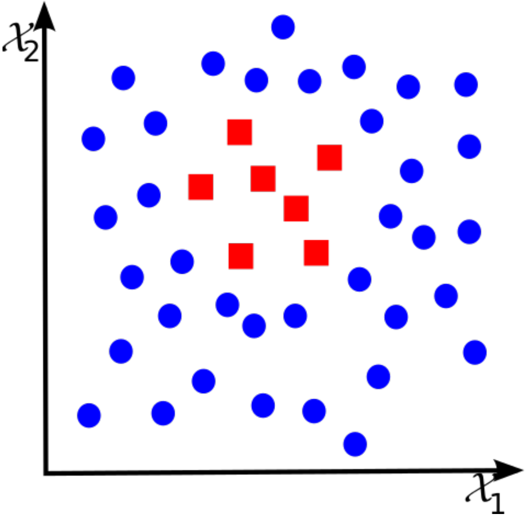
Feature Engineering

Strategies to cope with common challenges

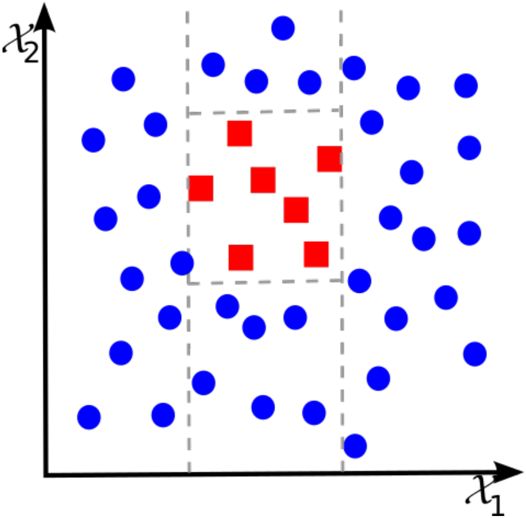
Principle Component Analysis

Kernel methods

Strategies to cope with non-linear problems

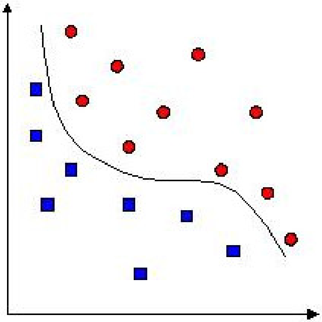


Strategies to cope with non-linear problems



Strategies to cope with non-linear problems

Classifier may search an objective function of sufficient dimension

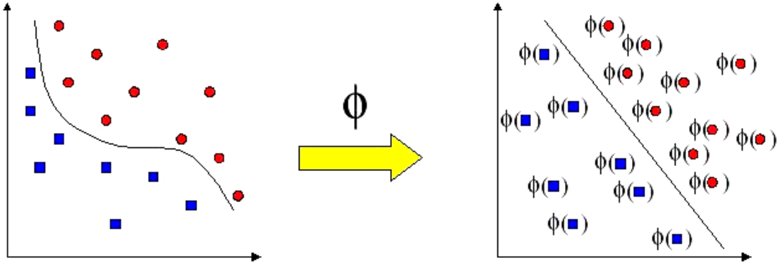


Strategies to cope with non-linear problems

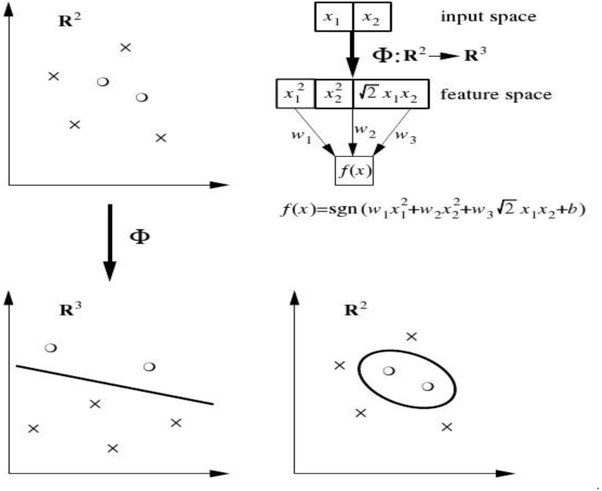
Alternative

Classifier may search an objective function of sufficient dimension for complex non-linear decision boundaries:

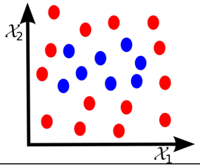
Change dimension of input space so that linear separation is possible



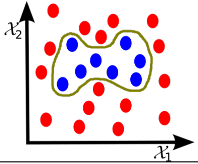
Example: Mapping into linear separable space



Using a kernel function

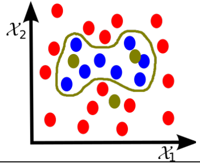


Using a kernel function



Hypothesis $= 1$ if

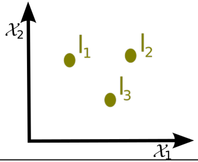
Using a kernel function



Hypothesis $= 1$ if

$$\Rightarrow w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \cdots \geq 0$$

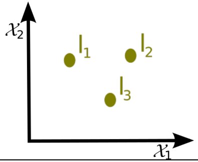
Using a kernel function



$$\Rightarrow \quad w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$$

Kernel Define kernel via landmarks

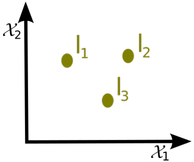
Using a kernel function



$$\Rightarrow w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \dots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x - l_i||^2}{2\sigma^2}}$

Using a kernel function

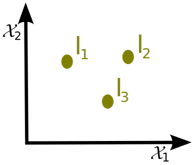


$$\Rightarrow \quad w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \ldots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x-l_i||^2}{2\sigma^2}}$

$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (towards 0 else)

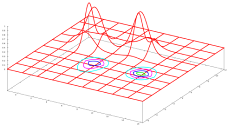
Using a kernel function



$$\Rightarrow \quad w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \dots$$

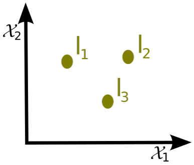
$$\text{Gaussian:} \quad k(x, l_i) = e^{-\frac{||x - l_i||^2}{2\sigma^2}}$$

$$x \approx l_i \quad \Rightarrow k(x, l_i) \approx 1 \text{ (towards 0 else)}$$



$\sigma = 1$

Using a kernel function



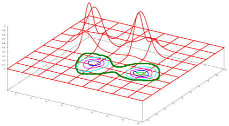
$$\Rightarrow \quad w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \dots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x - l_i||^2}{2\sigma^2}}$

$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (towards 0 else)

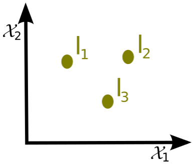
Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$



$\sigma = 1$

Using a kernel function



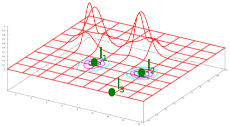
$$\Rightarrow \; w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \dots$$

Gaussian: $k(x, l_i) = e^{-\frac{||x - l_i||^2}{2\sigma^2}}$

$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (towards 0 else)

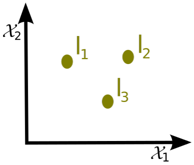
Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$



$\sigma = 1$

Using a kernel function



$\Rightarrow \quad w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \dots$

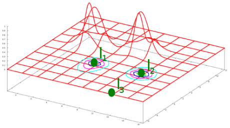
Gaussian: $k(x, l_i) = e^{-\frac{||x - l_i||^2}{2\sigma^2}}$

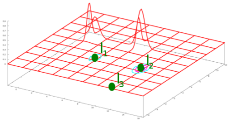
$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (towards 0 else)

σ controls the width of the Gaussian

Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

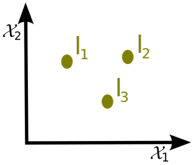
$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$





$\sigma = 1$ $\qquad$ $\sigma = 0.5$

Using a kernel function



$$\Rightarrow \quad w_0 + w_1 k_1 + w_2 k_2 + w_3 k_3 + \dots$$

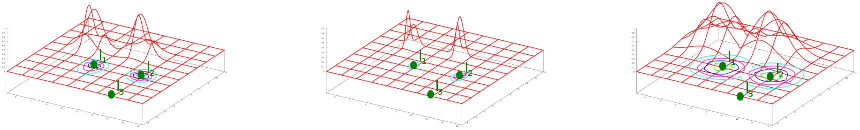
Gaussian: $k(x, l_i) = e^{-\frac{||x-l_i||^2}{2\sigma^2}}$

$x \approx l_i \Rightarrow k(x, l_i) \approx 1$ (towards 0 else)

σ controls the width of the Gaussian

Example: $w_0 = -0.5, w_1 = 1, w_2 = 1, w_3 = 0$

$$h(x) = w_0 + w_1 k(x, l_1) + w_2 k(x, l_2) + w_3 k(x, l_3)$$



$\sigma = 1$ $\quad$ $\sigma = 0.5$ $\quad$ $\sigma = 2$

Using a kernel function

Kernels – placement of landmarks

Possible choice of initial landmarks: All training-set samples

Training of w_i

$$f_i = \begin{bmatrix} k(x_i, l_1) \\ \vdots \\ k(x_i, l_m) \end{bmatrix}$$

$$\min_W C \sum_{i=1}^{m} y_i \text{cost}_{y_i=1}(W^T f_i) + (1 - y_i) \cdot \text{cost}_{y_i=0}(W^T f_i) + \frac{1}{2} \sum_{j=1}^{m} w_j^2$$

Questions?

Dariush Salami
dariush.salami@aalto.fi

Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.