Aalto University
School of Electrical Engineering

CS-C3240 – Machine Learning D

Round 3: From features to classification

Stephan Sigg

Department of Communications and Networking
Aalto University, School of Electrical Engineering
stephan.sigg@aalto.fi

Version 2.3, January 23, 2022

Outline

Feature Engineering

Feature engineering

Example: Voiced vs. unvoiced audio

A way to detect voice in audio is to calculate the number of zero-crossing. A 100 Hz signal will cross zero 100 times per second; an unvoiced segments can have 3000 zero crossing per second.

Feature pre-processing

- → Domain knowledge available?
- → Normalisation
- → Overlapping windows
- → Detection of outliers
- → Are features independent?

Feature engineering

Feature pre-processing

- → Domain knowledge available?
- → Normalisation
- → Overlapping windows
- → 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

→ Domain knowledge available?
→ Normalisation
→ Overlapping windows
→ 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

- → Domain knowledge available?
- → Normalisation
- → Overlapping windows
- → 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

- → Domain knowledge available?
- → Normalisation
- → Overlapping windows
- → 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

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

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

Feature pre-processing

→ Domain knowledge available?
→ Normalisation
→ Overlapping windows
→ Detection of outliers
→ Are features independent?

Feature engineering

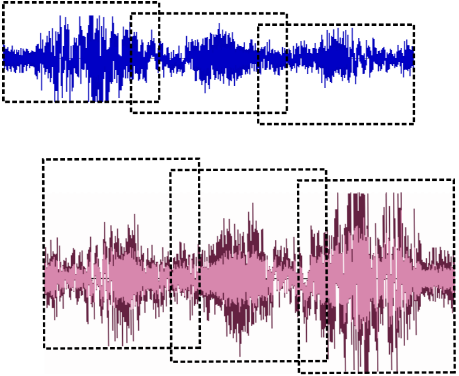
Important:

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

- → Domain knowledge available?
- → Normalisation
- → Overlapping windows
- → Detection of outliers
- → Are features independent?

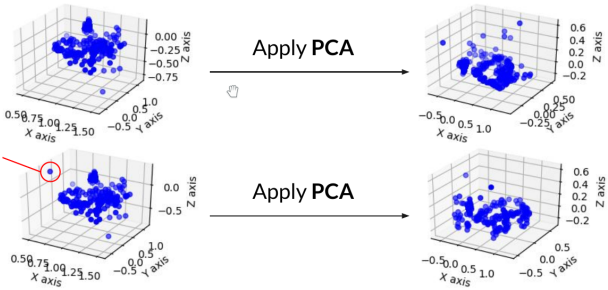
Feature engineering



Feature pre-processing

→ Domain knowledge available?
→ Normalisation
→ Overlapping windows
→ Detection of outliers
→ Are features independent?

Feature engineering



Feature pre-processing

→ Domain knowledge available?
→ Normalisation
→ Overlapping windows
→ 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

→ Domain knowledge available?

→ Normalisation

→ Overlapping windows

→ 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 not throw away outliers, unless you have evidence that they are errors

Darell Huff, How to lie with Statistics, 1954

Feature pre-processing

→ Domain knowledge available?
→ Normalisation
→ Overlapping windows
→ **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

- $\rightarrow$ Domain knowledge available?
- $\rightarrow$ Normalisation
- $\rightarrow$ Overlapping windows
- $\rightarrow$ 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, one-class SVM

Feature pre-processing

$\rightarrow$ Domain knowledge available?

$\rightarrow$ Normalisation

$\rightarrow$ Overlapping windows

$\rightarrow$ Detection of outliers

$\rightarrow$ Are features independent?

Feature engineering

Feature pre-processing

→ Domain knowledge available?
→ Normalisation
→ Overlapping windows
→ Detection of outliers
→ Are features independent?

Feature engineering

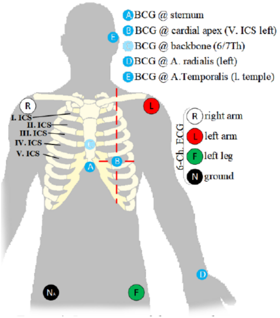
Examples for dependent features:

Feature pre-processing

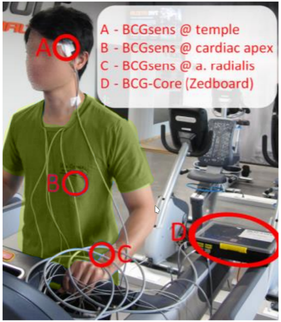
- → Domain knowledge available?
- → Normalisation
- → Overlapping windows
- → Detection of outliers
- → Are features independent?

Feature engineering

Example: walking speed vs. heart rate



(a) Positioning of the sensors



(b) Subject performing the study

Feature pre-processing

- → Domain knowledge available?
- → Normalisation
- → Overlapping windows
- → Detection of outliers
- → Are features independent?

Feature Selection

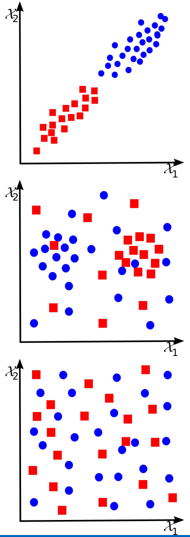
A large portion of the performance of Machine Learning algorithms is due to the right choice and processing of features.

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

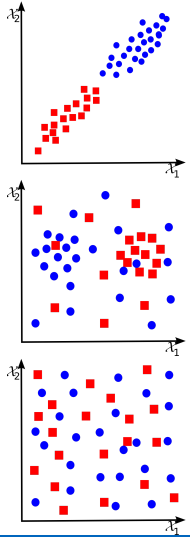
A large portion of the performance of Machine Learning algorithms is due to the right choice and processing of features.

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 $\mathcal{C}_i \in \{\mathcal{C}\}$?

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 $\mathcal{C}_i \in \{\mathcal{C}\}$?

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 $\mathcal{C}_i \in \{\mathcal{C}\}$?

Focus algorithm

1 Train and evaluate a classifier for singleton feature $\mathcal{X}_o$

2 Evaluate each set of two features $\mathcal{X}_o, \mathcal{X}_p$

⋮

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 $\mathcal{C}_i \in \{\mathcal{C}\}$?

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 $\mathcal{C}_i \in \{\mathcal{C}\}$?

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 $\mathcal{C}_i \in \{\mathcal{C}\}$?

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 $\mathcal{C}_i \in \{\mathcal{C}\}$?

Pearson Correlation Coefficient

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

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

Feature selection algorithms

How to identify good/meaningful features?

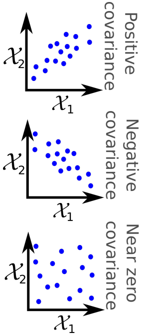
Feature selection

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 $\mathcal{C}_i \in \{\mathcal{C}\}$?

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?

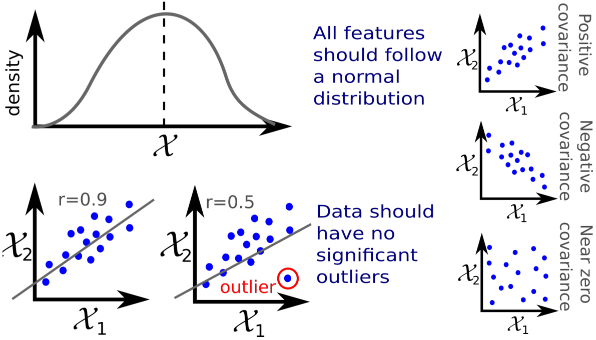
Feature selection

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 $\mathcal{C}_i \in \{\mathcal{C}\}$?

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?

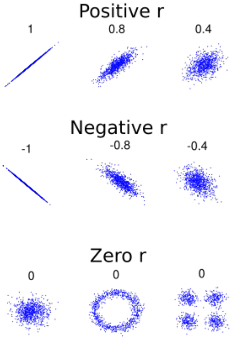
Feature selection

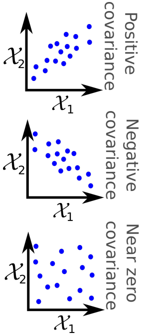
For a set of features $\{\mathcal{X}\}$, how to find a good subset $\{\mathcal{X}\}_s \subseteq \mathcal{X}\}$ which is best suited to distinguish between the considered classes $\mathcal{C}_i \in \{\mathcal{C}\}$?

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 $\mathcal{X}_i$





Variance and Covariance

For two features $\mathcal{X}_1, \mathcal{X}_2$, consider sets of measurements with zero mean:

$$\overrightarrow{x_1} = \{x_1^{(1)}, \dots, x_1^{(n)}\}$$
$$\overrightarrow{x_2} = \{x_2^{(1)}, \dots, x_2^{(n)}\}$$

Variance and Covariance

For two features $\mathcal{X}_1, \mathcal{X}_2$, consider sets of measurements with zero mean:

$$\overrightarrow{x_1} = \{x_1^{(1)}, \ldots, x_1^{(n)}\}$$
$$\overrightarrow{x_2} = \{x_2^{(1)}, \ldots, x_2^{(n)}\}$$

Variance: $E[(x_j^{(i)} - \mu)^2]$

Variance and Covariance

For two features $\mathcal{X}_1, \mathcal{X}_2$, consider sets of measurements with zero mean:

$$\overrightarrow{x_1} = \{x_1^{(1)}, \dots, x_1^{(n)}\}$$
$$\overrightarrow{x_2} = \{x_2^{(1)}, \dots, x_2^{(n)}\}$$

Variance: $E[(x_j^{(i)} - \mu)^2]$

$$\xrightarrow{\text{zero mean}} E[x_j^{(i)} \cdot x_j^{(i)}]$$

Variance and Covariance

For two features $\mathcal{X}_1, \mathcal{X}_2$, consider sets of measurements with zero mean:

$$\overrightarrow{x_1} = \{x_1^{(1)}, \ldots, x_1^{(n)}\}$$
$$\overrightarrow{x_2} = \{x_2^{(1)}, \ldots, x_2^{(n)}\}$$

Variance: $E[(x_j^{(i)} - \mu)^2]$

$\xrightarrow{\text{zero mean}} E[x_j^{(i)} \cdot x_j^{(i)}]$

$\Rightarrow \frac{1}{n} \sum_{i=1}^{n} (x_j^{(i)} \cdot x_j^{(i)})$

Variance and Covariance

For two features $\mathcal{X}_1, \mathcal{X}_2$, consider sets of measurements with zero mean:

$$\overrightarrow{x_1} = \{x_1^{(1)}, \ldots, x_1^{(n)}\}$$
$$\overrightarrow{x_2} = \{x_2^{(1)}, \ldots, x_2^{(n)}\}$$

Variance: $E[(x_j^{(i)} - \mu)^2]$

$\xrightarrow{\text{zero mean}} E[x_j^{(i)} \cdot x_j^{(i)}]$

$\Rightarrow \frac{1}{n} \sum_{i=1}^{n} (x_j^{(i)} \cdot x_j^{(i)})$

Covariance: $\frac{1}{n} \sum_{i=1}^{n} (x_1^{(i)} \cdot x_2^{(i)})$

Variance and Covariance

For two features $\mathcal{X}_1, \mathcal{X}_2$, consider sets of measurements with zero mean:

$$\overrightarrow{x_1} = \{x_1^{(1)}, \ldots, x_1^{(n)}\}$$
$$\overrightarrow{x_2} = \{x_2^{(1)}, \ldots, x_2^{(n)}\}$$

Variance: $E[(x_j^{(i)} - \mu)^2]$

$\xrightarrow{\text{zero mean}} E[x_j^{(i)} \cdot x_j^{(i)}]$

$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}(x_j^{(i)} \cdot x_j^{(i)})$

Covariance: $\frac{1}{n}\sum_{i=1}^{n}(x_1^{(i)} \cdot x_2^{(i)})$

$\Rightarrow \frac{1}{n}\overrightarrow{x_1}\,\overrightarrow{x_2}^T$

Covariance matrix

covariance: $\frac{1}{n}\sum_{i=1}^{n}\left(x_1^{(i)} \cdot x_2^{(i)}\right)$

$\Rightarrow \frac{1}{n}\overrightarrow{x_1}\,\overrightarrow{x_2}^{T}$

Covariance matrix

covariance: $\frac{1}{n}\sum_{i=1}^{n}(x_1^{(i)} \cdot x_2^{(i)})$

$\Rightarrow \frac{1}{n}\overrightarrow{x_1}\overrightarrow{x_2}^T$

Feature matrix:

$$X = \begin{bmatrix} \overrightarrow{x_1} \\ \vdots \\ \overrightarrow{x_m} \end{bmatrix}$$

Covariance matrix

covariance: $\frac{1}{n}\sum_{i=1}^{n}(x_1^{(i)} \cdot x_2^{(i)})$

$\Rightarrow$ $\frac{1}{n}\overrightarrow{x_1}\overrightarrow{x_2}^T$

Feature matrix:

$$X = \begin{bmatrix} \overrightarrow{x_1} \\ \vdots \\ \overrightarrow{x_m} \end{bmatrix}$$

Covariance matrix:

$$\Sigma = \frac{1}{n}XX^T$$

Covariance matrix

covariance: $\frac{1}{n}\sum_{i=1}^{n}(x_1^{(i)} \cdot x_2^{(i)})$

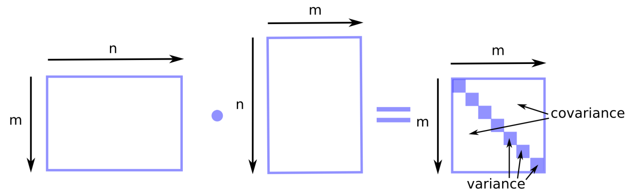
$\Rightarrow \frac{1}{n}\vec{x_1}\vec{x_2}^T$

Feature matrix:

$$X = \begin{bmatrix} \vec{x_1} \\ \vdots \\ \vec{x_m} \end{bmatrix}$$

Covariance matrix:

$$\Sigma = \frac{1}{n}XX^T$$



Questions?

Stephan Sigg
stephan.sigg@aalto.fi

Si Zuo
si.zuo@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.