Aalto University
School of Electrical
Engineering

Machine Learning for pervasive systems

Introduction

Stephan Sigg

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

Version 1.0, January 21, 2022

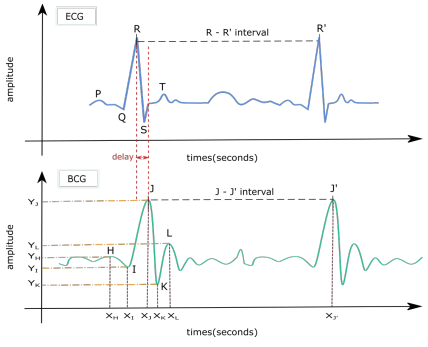
Machine learning in Pervasive Systems

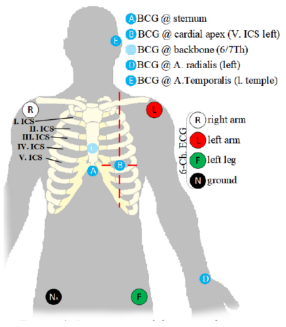
Pervasive Computing

Pervasive computing goes past the arena of desktops so that virtually any device, from apparel to kitchen appliances, could be embedded with microchips, connecting these devices to a boundless network of other gadgets.

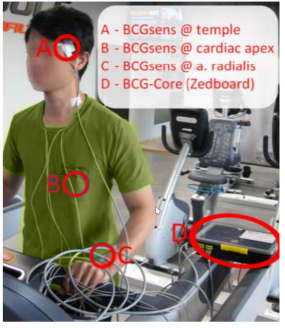
PerCom example

Example: Machine learning in Pervasive Systems



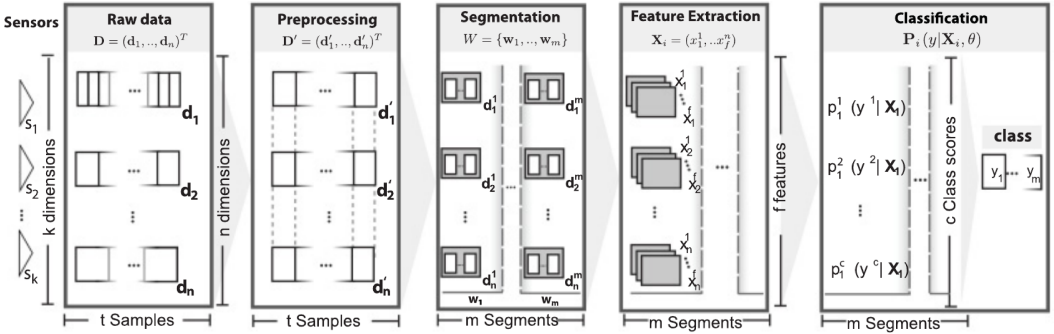


(a) Positioning of the sensors



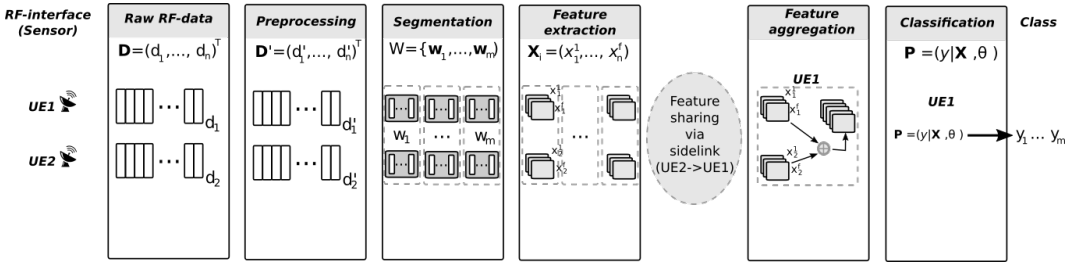
(b) Subject performing the study

Toolchain for ML in Pervasive Systems

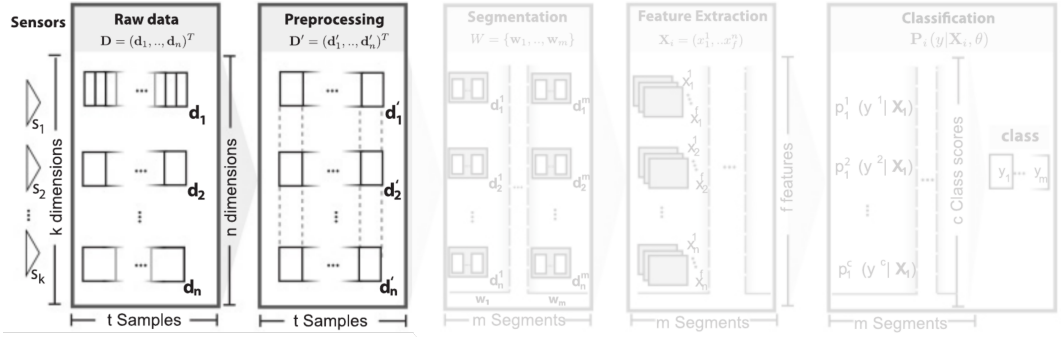


Bulling, Blanke & Schiele, ACM Computing Surveys, vol. 46, no. 3, January 2014

Toolchain for ML in Pervasive Systems



Pre-processing

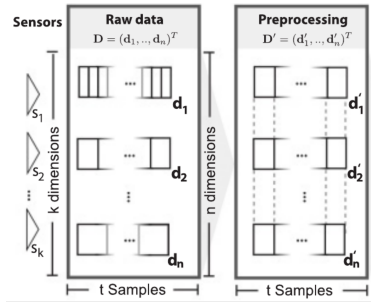


Bulling, Blanke & Schiele, ACM Computing Surveys, vol. 46, no. 3, January 2014

Pre-processing

Need for pre-processing:

- sample frequency might differ
- sample frequency might change (e.g. power saving)
- non-synchronized devices
- corrupted data (e.g. physical activity, malfunction)
- electromagnetic interference with AC power lines (EEG, EMG, EOG,...)



Pre-processing is to synchronize data streams and to remove artifacts.

Example: Pre-processing

raw BCG signal

↓

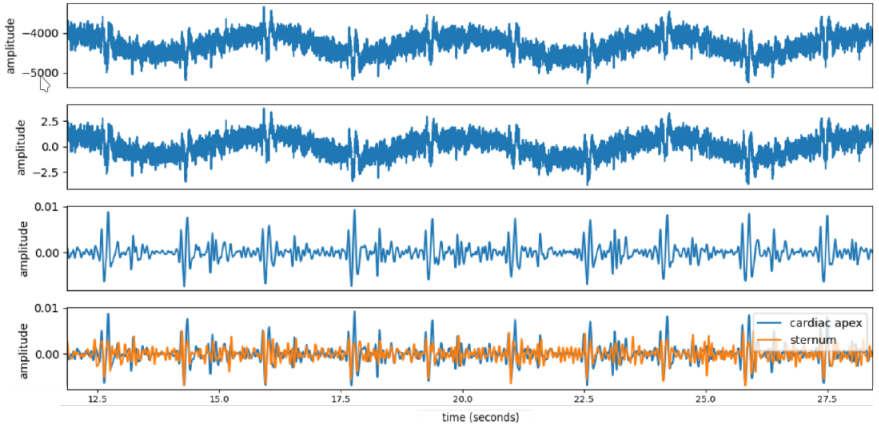
normalization

↓

band-pass filtering and differencing

↓

synchronization signals



Pre-processing – Exploratory data analysis

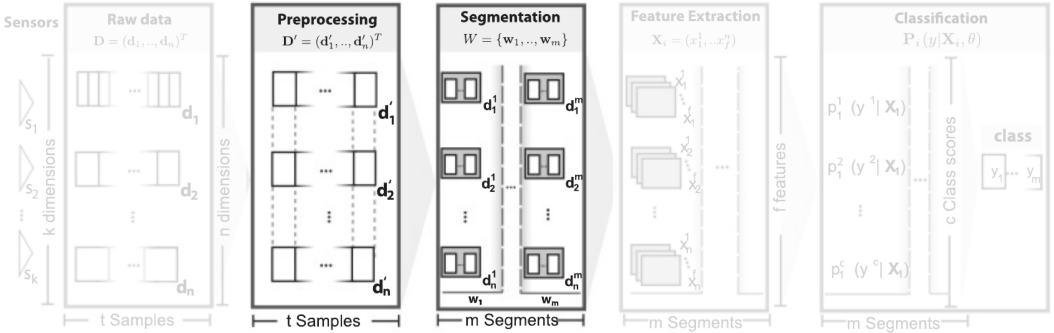
Steps to apply before feature engineering:[1]

- data with low variance tends to have little explanatory power (ingore)
- test for randomness (standard randomness tests). Random data has no explanatory power and can be discarded.
- find and handle outliers
- find and handle missing values
- scatterplots help visualizing correlations between data
- check/verify that the data follows expected distributions (if any) (e.g. using histograms or Skewness). This may help to discover potential data extraction errors or hidden biases in the sampling.

Useful tool for exploratory data analysis: https://openrefine.org/

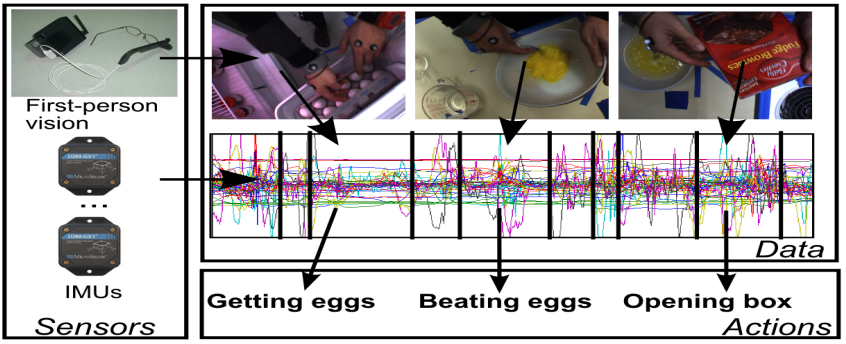
[1] Further reading: Glenn Myatt and Wayne Johnson. Making sense of data i: A practical guide to exploratory data analysis and data mining, 2014.

Segmentation



Bulling, Blanke & Schiele, ACM Computing Surveys, vol. 46, no. 3, January 2014

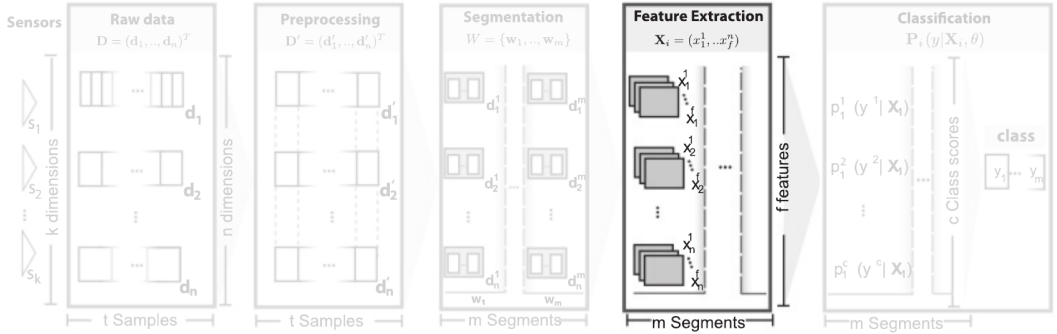
Data segmentation



- spotting parts of the data which correspond to relevant activities
- reduces processing load and energy consumption by disregarding part of the data

Exact boundaries of an activity are often difficult to define.

Features extraction



Bulling, Blanke & Schiele, ACM Computing Surveys, vol. 46, no. 3, January 2014

Problem Often, raw data not well suited as input for Machine learning approaches

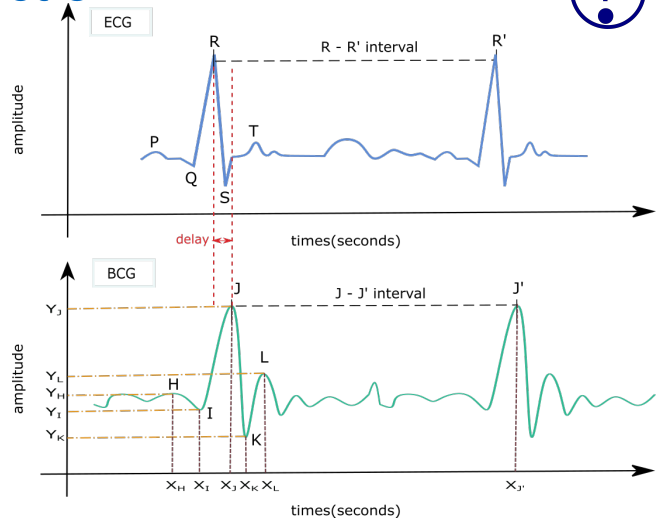
→ Noisy

→ Not meaningful/expressive

Feature extraction Meaningful representation and correlated to respective classes

Example: Feature extraction

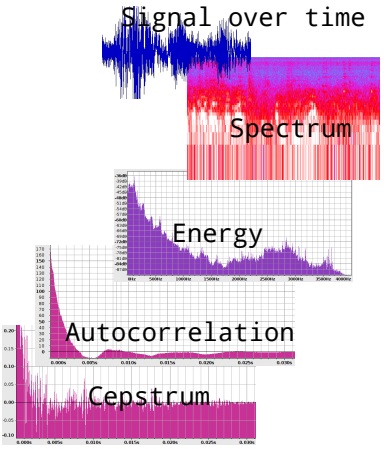
Features:



Features and feature extraction

What is a feature?

- → Loudness
- → Energy on frequency bands
- → Zero crossings
- → Direction changes
- → Mean and variance of the signal
- → Zero Crossing Rate
- → Mean Crossing rate
- → Cepstral coefficients
- → Spectral entropy
- → Energy in Frequency bands
- → Min, max, ...



Feature extraction

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 extraction

Feature pre-processing

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

Feature extraction

Simple normalization: Scaling

For each sample x_i from a data set X, compute the scaled value as

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

Feature pre-processing

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

Feature extraction

Simple normalization: Scaling

For each sample x_i from a data set X, compute the scaled value as

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

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 extraction

Standardization to zero mean/unit variance

Given the mean μ_F and standard deviation σ_F for a feature F, for each feature sample x_i^F, the standardize feature sample is

$$x_i'^F = \frac{x_i^F - \mu_F}{\sigma_F}$$

Feature pre-processing

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

Feature extraction

Standardization to zero mean/unit variance

Given the mean μ_F and standard deviation σ_F for a feature F, for each feature sample x_i^F, the standardize feature sample is

$$x_i'^F = \frac{x_i^F - \mu_F}{\sigma_F}$$

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

Feature pre-processing

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

Feature extraction

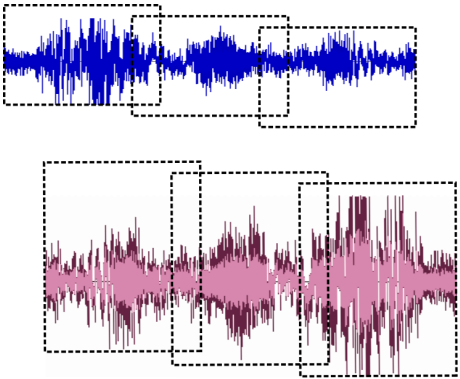
Important:

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?

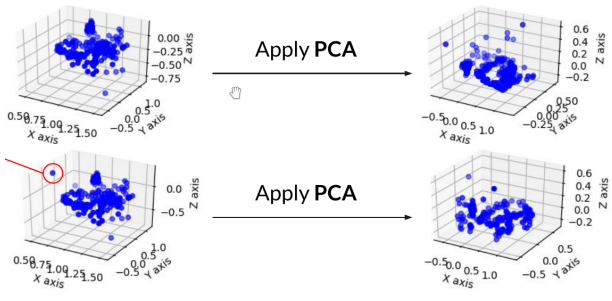
Feature extraction



Feature pre-processing

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

Feature extraction



Feature pre-processing

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

Feature extraction

Feature pre-processing

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

Feature extraction

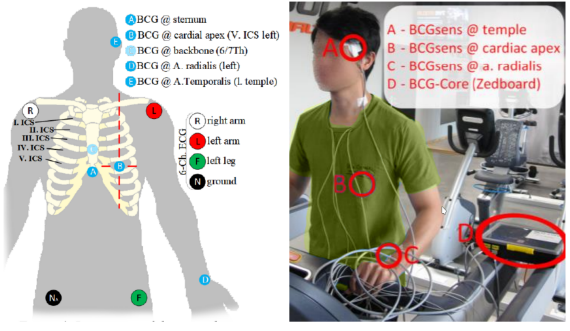
Examples for dependent features:

Feature pre-processing

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

Feature extraction

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.

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

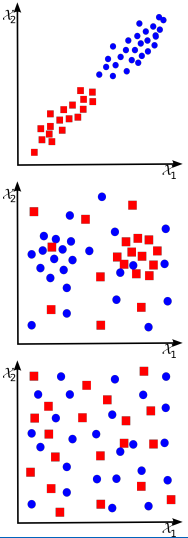
A large portion of the performance of Machine Learning algorithms is due to the right choice and processing of features.

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

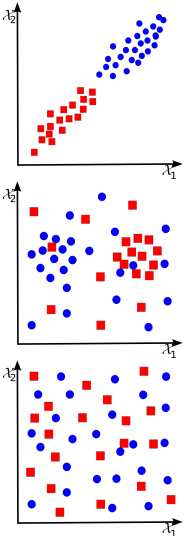
A large portion of the performance of Machine Learning algorithms is due to the right choice and processing of features.

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

How to find a good subset of features which is best suited to distinguish between the classes considered?

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

How to find a good subset of features which is best suited to distinguish between the classes considered?

Las Vegas Filter

Repeatedly generate random feature subsets and compute their classification performance

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

How to find a good subset of features which is best suited to distinguish between the classes considered?

Focus algorithm

1. Evaluate each singleton feature set
2. Evaluate each set of two features

⋮

Until consistent solution is found

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

How to find a good subset of features which is best suited to distinguish between the classes considered?

Focus algorithm

1. Evaluate each singleton feature set
2. Evaluate each set of two features

⋮

Until consistent solution is found

Complexity:

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

$$\binom{|\chi|}{1} \cdot \binom{|\chi|}{2} \cdots \binom{|\chi|}{|\chi|}$$

Feature selection algorithms

How to identify good/meaningful features?

Feature selection

How to find a good subset of features which is best suited to distinguish between the classes considered?

Relief algorithm

Given a single feature, compute for all samples

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

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

How to find a good subset of features which is best suited to distinguish between the classes considered?

Relief algorithm

Given a single feature, compute for all samples

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

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

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

How to find a good subset of features which is best suited to distinguish between the classes considered?

Pearson Correlation Coefficient

$$\varrho(\chi_1, \chi_2) = \frac{\mathrm{Cov}(\chi_1, \chi_2)}{\sqrt{\mathrm{Var}(\chi_1)\mathrm{Var}(\chi_2)}}$$

- Identifies linear relation between features χ_i

Feature selection algorithms

How to identify good/meaningful features?

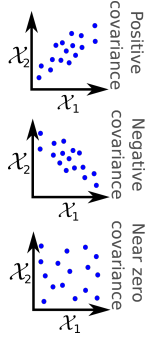
Feature selection

How to find a good subset of features which is best suited to distinguish between the classes considered?

Pearson Correlation Coefficient

$$\varrho(\chi_1, \chi_2) = \frac{\mathrm{Cov}(\chi_1, \chi_2)}{\sqrt{\mathrm{Var}(\chi_1)\mathrm{Var}(\chi_2)}}$$

- Identifies linear relation between features χ_i



Feature selection algorithms

How to identify good/meaningful features?

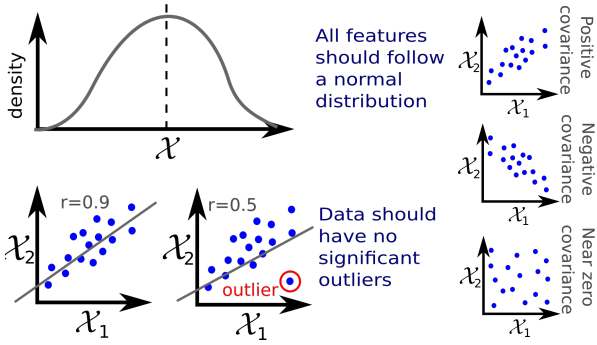
Feature selection

How to find a good subset of features which is best suited to distinguish between the classes considered?

Pearson Correlation Coefficient

$$\varrho(\chi_1, \chi_2) = \frac{\mathrm{Cov}(\chi_1, \chi_2)}{\sqrt{\mathrm{Var}(\chi_1)\mathrm{Var}(\chi_2)}}$$

- Identifies linear relation between features χ_i



Feature selection algorithms

How to identify good/meaningful features?

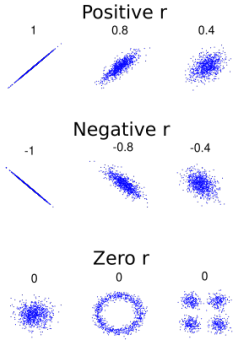
Feature selection

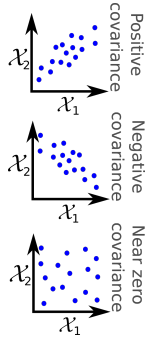
How to find a good subset of features which is best suited to distinguish between the classes considered?

Pearson Correlation Coefficient

$$\varrho(\chi_1, \chi_2) = \frac{\mathrm{Cov}(\chi_1, \chi_2)}{\sqrt{\mathrm{Var}(\chi_1)\mathrm{Var}(\chi_2)}}$$

- Identifies linear relation between features χ_i





Feature Selection

Linear Discriminant Analysis

Find linear combination of features to characterize or separate classes of a categorical variable.

Anova

ANOVA (Analysis of variance) is a statistical test of whether the means of several groups are equal.

Chi-square

Statistical test for groups of categorical features to evaluate likelihood of correlation between them using their frequency distribution.

Feature Selection

Linear Discriminant Analysis

Find linear combination of features to characterize or separate classes of a categorical variable.

Anova

ANOVA (Analysis of variance) is a statistical test of whether the means of several groups are equal.

Chi-square

Statistical test for groups of categorical features to evaluate likelihood of correlation between them using their frequency distribution.

Feature Selection

Linear Discriminant Analysis

Find linear combination of features to characterize or separate classes of a categorical variable.

Anova

ANOVA (Analysis of variance) is a statistical test of whether the means of several groups are equal.

Chi-square

Statistical test for groups of categorical features to evaluate likelihood of correlation between them using their frequency distribution.

Feature Selection algorithms

Wrapper methods (random search)

- Train a model on a subset of features
- Add or remove features based on the result

Forward selection

Iteratively add feature that best improves model performance until no further improvement

Backward elimination

Iteratively remove least significant feature until no further improvement achieved.

Recursive feature elimination

- Repeatedly create models and compute best and worst performing feature in each iteration.
- Recursive with left-over features.
- Features ranked based on order of elimination.

Feature Selection algorithms

Wrapper methods (random search)

- Train a model on a subset of features
- Add or remove features based on the result

Forward selection

Iteratively add feature that best improves model performance until no further improvement

Backward elimination

Iteratively remove least significant feature until no further improvement achieved.

Recursive feature elimination

- Repeatedly create models and compute best and worst performing feature in each iteration.
- Recursive with left-over features.
- Features ranked based on order of elimination.

Feature Selection algorithms

Wrapper methods (random search)

- Train a model on a subset of features
- Add or remove features based on the result

Forward selection

Iteratively add feature that best improves model performance until no further improvement

Backward elimination

Iteratively remove least significant feature until no further improvement achieved.

Recursive feature elimination

- Repeatedly create models and compute best and worst performing feature in each iteration.
- Recursive with left-over features.
- Features ranked based on order of elimination.

Feature Selection algorithms

Wrapper methods (random search)

- Train a model on a subset of features
- Add or remove features based on the result

Forward selection

Iteratively add feature that best improves model performance until no further improvement

Backward elimination

Iteratively remove least significant feature until no further improvement achieved.

Recursive feature elimination

- Repeatedly create models and compute best and worst performing feature in each iteration.
- Recursive with left-over features.
- Features ranked based on order of elimination.

Feature Selection

Embedded methods

LASSO

Controls feature weights via L1 regularization
(penalty equivalent to the absolute value of the magnitude of the coefficients)

RIDGE

Controls feature weights via L2 regularization
(penalty equivalent to the square of the magnitude of the coefficients)

Feature Selection

Embedded methods

LASSO

Controls feature weights via L1 regularization
(penalty equivalent to the absolute value of the magnitude of the coefficients)

RIDGE

Controls feature weights via L2 regularization
(penalty equivalent to the square of the magnitude of the coefficients)

Feature Selection

Embedded methods

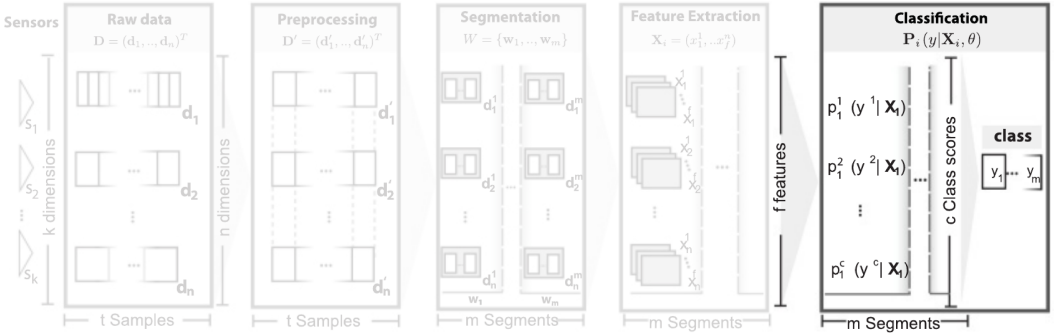
LASSO

Controls feature weights via L1 regularization
(penalty equivalent to the absolute value of the magnitude of the coefficients)

RIDGE

Controls feature weights via L2 regularization
(penalty equivalent to the square of the magnitude of the coefficients)

Classification



Bulling, Blanke & Schiele, ACM Computing Surveys, vol. 46, no. 3, January 2014

Example: Machine Learning in Pervasive Systems

Pantomime video

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.