Machine Learning Riassunto Completo Carlo Vercellis

Machine Learning (Politecnico di Milano)

Scansiona per aprire su Studocu

Studocu non è sponsorizzato o supportato da nessuna università o ateneo.
Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

Machine Learning

Table of Contents

0. Introduction 7

1. Data Preparation 8

1.1 Incomplete Data/Missing Values 8

1.1.1 Elimination 8

1.1.2 Inspection 8

1.1.3 Identification 8

1.1.4 Replacement 8

1.1.4.1 Mean Value of the Attribute 8

1.1.4.2 Mean Value of the Attribute for the same Target Class 8

1.1.4.3 Value Estimated using Statistical Inference 8

1.2 Noisy Data/Outliers Detection 8

1.2.1 Central Limit Theorem 8

1.2.2 Tchebysheff Theorem 9

1.2.3 Z-Index 9

1.2.4 Clustering 9

1.2.5 Correcting Outliers 9

1.3 Data Transformation/Feature Engineering 10

1.3.1 Data Standardization 10

1.3.1.1 Decimal Scaling 10

1.3.1.2 Min Max Scaling 10

1.3.1.3 Standard Scaling 10

1.3.2 Data Reduction 10

1.3.2.1 Data Sampling 10

1.3.2.1.1 Pure Random Sampling 10

1.3.2.1.2 Stratified Sampling 10

1.3.2.2 Attribute Selection or Projection 11

1.3.2.2.1 Attribute Selection 11

1.3.2.2.1.1 Filtering 11

1.3.2.2.1.2 Wrapper Methods 11

1.3.2.2.1.3 Embedded Methods 11

1.3.2.2.2 Projection (PCA) 11

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

1.3.2.3 Discretization or Aggregation 12
1.3.2.3.1 Discretization 12
1.3.2.3.2 Aggregation 12
2. Exploratory Data Analysis 13
2.1 Uni-Variate Analysis 13
2.1.1 Categorical Attributes 13
2.1.1.1 Graphical Analysis 13
2.1.1.2 Heterogeneity Index 13
2.1.2 Numerical Attributes 13
2.1.2.1 Graphical Analysis 13
2.1.2.2 Measures of Central Tendency 14
2.1.2.3 Measures of Dispersion 14
2.1.2.4 Measures of Relative Location 14
2.1.2.5 Analysis of the Empirical Density 15
2.1.2.5.1 Box and Whiskers 15
2.1.2.5.2 Asymmetry Index 15
2.1.2.5.3 Normality 15
2.1.2.5.3.1 Kurtis Index 15
2.1.2.5.3.2 Normal Probability Plot 15
2.2 Bi-Variate Analysis 16
2.2.1 Graphical Analysis 16
2.2.1.1 Scatter Plot 16
2.2.1.2 Loess Diagram 16
2.2.1.3 Contour Lines 16
2.2.1.4 Quantile-Quantile Plot 16
2.2.1.5 Box and Whiskers 16
2.2.2 Correlation Measures 17
2.2.2.1 Contingency Tables 17
2.3 Multivariate Analysis 17
2.3.1 Scatter Plot 17
2.3.2 Star Plot 18
2.3.3 Spider Web Charts 18
3. Classification 19
3.1 Key Concepts 19
3.1.1 Majority Vote Principle 19
3.1.2 Classification as a Learning Task 19

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.1.2.1 Data Format 19
3.1.2.2 Application Domain 19
3.1.2.3 Mathematical Formulation 19
3.1.3 Multi-Class Approaches 20
3.1.3.1 One Against All 20
3.1.3.2 All Against All 20
3.1.4 Taxonomy of Classification 21
3.1.5 Assessing Classification Methods 21
3.1.5.1 Prediction Accuracy 22
3.1.5.1.1 Indicators 22
3.1.5.1.1.1 Loss Function 22
3.1.5.1.1.2 Empirical Error 22
3.1.5.1.2 Splitting Methods (Train and Test Sets) 22
3.1.5.1.2.1 Old-Out Method 22
3.1.5.1.2.2 Repeated Random Sampling Method 22
3.1.5.1.2.3 K-fold Cross Validation Method 23
3.1.5.1.2.4 Leave one Out Method 23
3.1.5.1.3 Advanced Accuracy Indicators 23
3.1.5.1.3.1 Confusion Matrix 23
3.1.5.1.3.2 ROC Curve 24
3.1.5.1.3.3 Cumulative Gain 24
3.1.5.1.3.4 Lift 24
3.1.6 Overfitting 25
3.1.7 Numerical representation of Categorical variables 25
3.2 K-Nearest Neighbors 25
3.3 Classification Tree 26
3.3.1 General Introduction 26
3.3.1.1 Characteristics of Classification Trees 26
3.3.1.2 Pre-Pruning (Stopping Rules) 26
3.3.1.2.1 Purity threshold 26
3.3.1.2.2 Minimum Cardinality 26
3.3.1.2.3 Minimum Information Gain 27
3.3.1.3 Post Pruning 27
3.3.1.4 Taxonomy of Classification Trees 27
3.3.1.4.1 Binary Vs General Trees 27
3.3.1.4.2 Univariate Vs Multivariate Trees 27

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.3.2 Mathematical Formulation (axis parallel only) 27
3.3.2.1 Measures of heterogeneity 28
3.3.2.2 Splitting Rule 28
3.3.2.2.1 Impurity Index: 28
3.3.2.2.2 Information Gain 28
3.3.3 Hyper parameters 28
3.4 Bayesian Methods 29
3.4.1 Naïve Bayesian Methods 29
3.4.2 Bayesian Belief Networks 30
3.5 Logistic Regression 30
3.6 Support Vector Machines 31
3.6.1 Accuracy Vs. Generalization 31
3.6.2 Classification and Statistical Learning Theory 31
3.6.3 SVM Formulation 32
3.6.3.1 Dog and Sheep Example 32
3.6.3.2 Linear Separation 33
3.6.3.2.1 Geometrical Interpretation 33
3.6.3.2.2 Hard Separation Problem 33
3.6.3.2.3 Relaxing the Problem for Outliers 33
3.6.3.2.4 How to solve the optimization problem: Lagrangean Relaxation and Duality ... 34
3.6.3.2.5 Predictions 34
3.6.3.3 Non-Linear Separation with SVM 34
3.6.3.3.1 Main Idea 34
3.6.3.3.2 Kernels 35
3.6.3.3.3 Predictions 35
3.6.3.4 Support Vectors: Complementarity of Slackness 35
3.6.3.5 Hyperparameters 35
3.7 Artificial Neural Networks 36
3.7.1 Perceptron 36
3.7.2 Multi-Layer Neural Networks 37
3.7.3 Non-Linear Separation 37
3.7.4 Deep Learning 38
3.8 Ensemble Methods 38
3.8.1 Bagging 39
3.8.2 Boosting 39
3.8.3 Bagging: Random Forest 39

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.8.4 Boosting: AdaBoost 39
4. Regression 40
4.1 Structure of Regression Problems 40
4.1.1 Spurious Correlation 40
4.1.2 Hypothesis Space 40
4.2 Least Square Regression 40
4.2.1 Simple Regression Problem 40
4.2.2 Multiple Linear regression 41
4.3 Assumptions on the Residuals 42
4.3.1 Normal, Zero Mean, Constant Variance 42
4.3.2 Independence of the Residuals 42
4.3.3 Heteroscedasticity 43
4.4 Ridge Regression 43
4.5 Lasso regression 43
4.6 Generalized Linear Models 43
4.7 K-Nearest Neighbors 44
4.8 Regression Trees 44
4.9 Support Vector Regression 44
4.10 Neural Network 44
4.11 Assessing Performance of the Model 45
4.11.1 Normality and Independence of the Residuals 45
4.11.1.1 Residuals VS Fitted 45
4.11.1.2 Standardized Residuals VS Fitted 45
4.11.1.3 QQ-Plot 45
4.11.2 Significance of the Coefficients 46
4.11.2.1 T-Student Test 46
4.11.2.2 ANOVA Test 46
4.11.3 Determination Coefficient 47
4.11.4 Regression Coefficient 47
4.11.5 Linear Correlation Coefficient 47
4.12 Multi-Collinearity of the Independent Variables 47
4.13 Confidence and Prediction Limits 48
5. Clustering 49
5.1 General Concepts 49
5.1.1 Taxonomy of Clustering Methods 49
5.1.2 Assessing Performance 50

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

5.1.3 Affinity Measures 50
5.1.4 Distance Measures 50
5.1.5 Affinity Measures for Binary Attributes 51
5.1.6 Affinity Measures: General Case 51
5.2 Partition Methods 52
5.2.1 K-Means Algorithm 52
5.2.1.1. Strengths and Weaknesses 52
5.2.2 K-Medoids Algorithm 53
5.3 Hyerarchical Methods 54
5.3.1 Agglomerative Methods 54
5.3.1.1 Measures of Distance Between Clusters 54
5.4 Clustering Validity 56
5.4.1 Number of Clusters 56
5.4.2 Internal Measures 56
6. Association Rules 57
6.1 Metrics for Assessing Associative Rules 58
6.1.1 Lift Index 58
6.2 Solution Algorithm 59

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

0. Introduction

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

1. Data Preparation

1.1 Incomplete Data/Missing Values

1.1.1 Elimination

It is always OK. Eliminate the entire row or, if many observations are NaN (more than 20%), the entire column. This could be the case of socio-demographic questions of a survey (non-mandatory ones).

1.1.2 Inspection

Manual inspection before eliminating or applying other techniques. Very costly from a human point of view, applicable only on small datasets or few NaNs.

1.1.3 Identification

Sometimes is OK. Applying a special value to all NaN observation, particularly useful with categorical variables or when NaNs correspond all to a specific situation (Missing value on wage=Unemployed).

1.1.4 Replacement

Not recommended at all, injection of random noise in your data.

1.1.4.1 Mean Value of the Attribute

1.1.4.2 Mean Value of the Attribute for the same Target Class

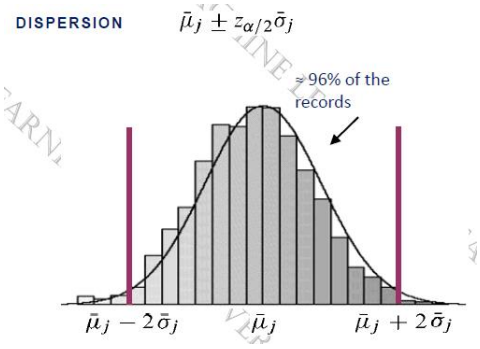
1.1.4.3 Value Estimated using Statistical Inference

1.2 Noisy Data/Outliers Detection

Refers only to numerical variables.

1.2.1 Central Limit Theorem

When data follows a normal distribution, 96% of observations fall within sample mean +/- 2 standard deviations.

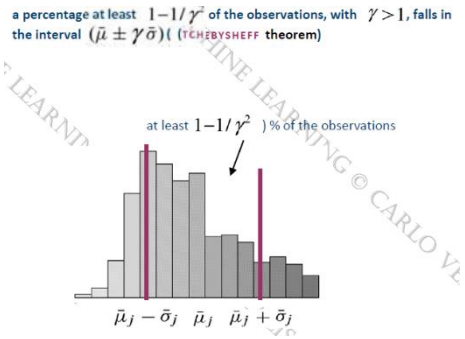


apply only if distribution is normal

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

1.2.2 Tchebysheff Theorem

For any distribution, 91% of observations fall within sample mean +/- 3 standard deviation.



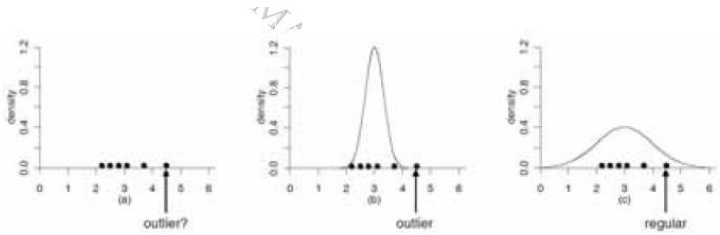
1.2.3 Z-Index

$$z_{ij}^{ind} = \frac{x_{ij} - \bar{\mu}_j}{\bar{\sigma}_j}$$

When Z-Index > 3 → It could be an outlier.

When Z-Index >> 3 → It is an outlier.

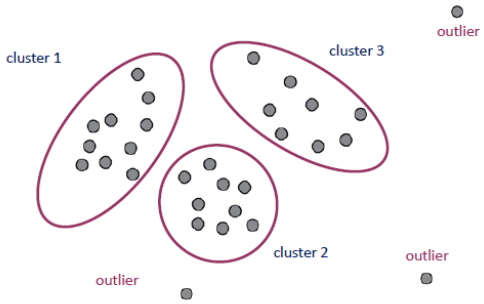
Mostly used for bell shaped distributions.



1.2.4 Clustering

Through cluster analysis we can identify outliers as observations that for a cluster by themselves, remaining out of any of the clusters created by the analysis.

Much more costly in computational time compared to simple Z-Index.

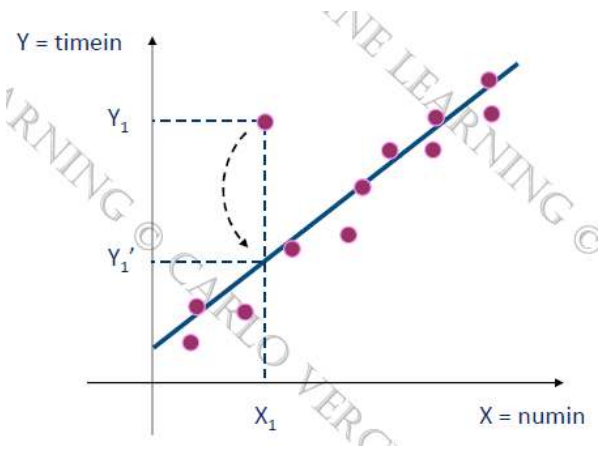


evaluate "density" in a neighborhood of a record: a record is an outlier if a percentage p of the records falls at a distance greater than d

1.2.5 Correcting Outliers

identification example

Perform this operation only when there is a KNOWN and ASSIGNABLE CAUSE for the outlier to be there. In this case the outlier's value can be changed performing a statistical inference (for example simple linear regression).



This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

1.3 Data Transformation/Feature Engineering

Simplifing the learning process for algorithms.

1.3.1 Data Standardization

DECIMAL SCALING $$x'_{ij} = \frac{x_{ij}}{10^h}$$

MIN-MAX method

$$x'_{ij} = \frac{x_{ij} - x_{min,j}}{x_{max,j} - x_{min,j}}(x'_{max,j} - x'_{min,j}) + x'_{min,j}$$

Z-INDEX method $$x'_{ij} = \frac{x_{ij} - \bar{\mu}_j}{\bar{\sigma}_j}$$

The goal is to have all the features on the same scale (same order of magnitude), of all numerical variables.

1.3.1.1 Decimal Scaling

Simplest standardization method. At the end you get values within 0 and 1.

1.3.1.2 Min Max Scaling

Slightly better than the previous. Usually you want to get values within 0 and 1.

1.3.1.3 Standard Scaling

If your data distribution is almost normal, you should get values within -3 and +3, if not you get pretty unpredictable values. In case of non-normal distributed data Min Max is preferrable.

1.3.2 Data Reduction

The goals are:

1) Improve Efficiency in Model Identification
2) Preserve Model Accuracy
3) Achieve Simpler Models

- **record reduction by SAMPLING**
- **attribute reduction by SELECTION or PROJECTION**
- **values reduction by DISCRETIZATION or AGGREGATION**

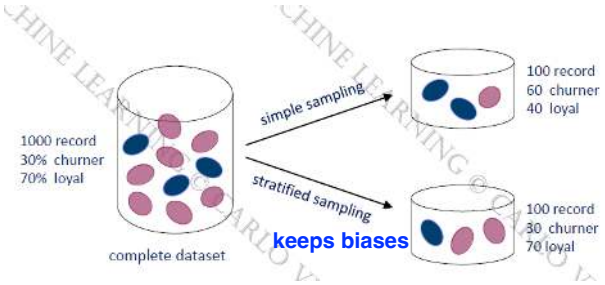
1.3.2.1 Data Sampling

You sample the rows to get a subset of the original observation (Row Elimination).

1.3.2.1.1 Pure Random Sampling

1.3.2.1.2 Stratified Sampling

(Stratified Sampling apply only to Classification Problems)



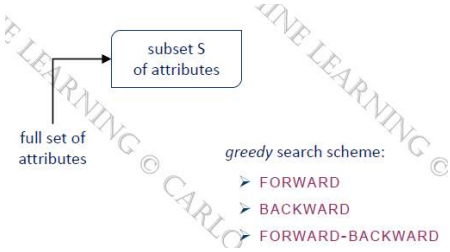
Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

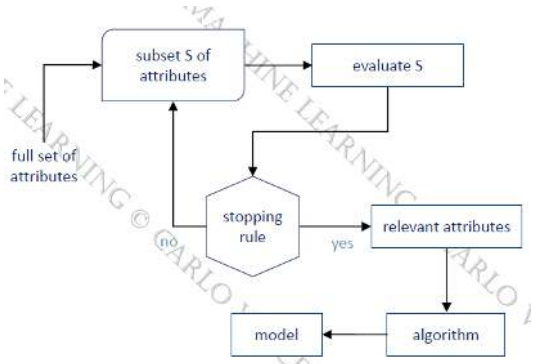
1.3.2.2 Attribute Selection or Projection

1.3.2.2.1 Attribute Selection

Select useful attributes and throw away some useless columns.

All methods follow a Greedy Logic: you cannot try all possible combinations of attributes, so you are never 100% sure your solution is optimal.

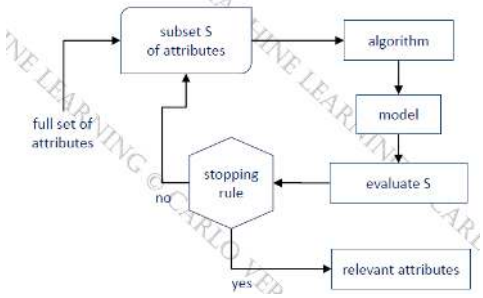




1.3.2.2.1.1 Filtering

You start from considering all the attributes and you evaluate them throwing away non-relevant ones. For example you could use correlation coefficient as an evaluation method in a regression problem.

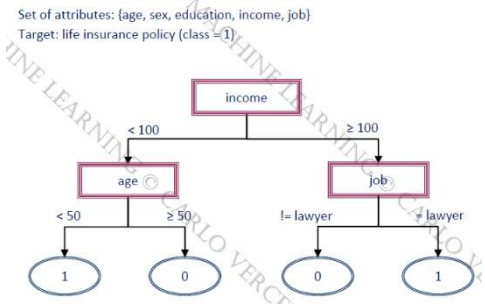
Very fast but not very accurate.



1.3.2.2.1.2 Wrapper Methods

You apply the algorithm and generate your model for each subset of attributes, until a stopping ruse is met, and you are satisfied with the results.

More accurate but since you have to apply the algorithm many times is way more costly. like forward/backward selectio



1.3.2.2.1.3 Embedded Methods

Attribute selection is directly embedded in the ML algorithm.

Best situation but applicable only to some specific algorithms (classification trees, random forests, SVM, artificial neural networks).

1.3.2.2.2 Projection (PCA)

$\mathbf{V} = \mathbf{X}'\mathbf{X}$ COVARIANCE matrix of data

$$\tilde{x}_{ij} = x_{ij} - \frac{1}{m}\sum_{i=1}^{m} x_{ij}.$$

$$\begin{aligned} E[\mathbf{w}_j'\mathbf{x}_i - E[\mathbf{w}_j'\mathbf{x}_i]]^2 &= E[(\mathbf{w}_j'(\mathbf{x}_i - E[\mathbf{x}_i]))^2] \\ &= \mathbf{w}_j' E[(\mathbf{x}_i - E[\mathbf{x}_i])'(\mathbf{x}_i - E[\mathbf{x}_i])]\mathbf{w}_j \\ &= \mathbf{w}_j'\mathbf{V}\mathbf{w}_j. \end{aligned}$$

The FIRST PRINCIPAL COMPONENT is obtained by solving an optimization problem

$$\max_{\mathbf{w}_1} \{ \mathbf{w}_1'\mathbf{V}\mathbf{w}_1 : \mathbf{w}_1'\mathbf{w}_1 = 1 \}$$

It can be shown that:

- the vector that solves the problem is the eigenvector associated to the largest eigenvalue
- the eigenvalue is equal to the amount of explained variance

The SECOND PRINCIPAL COMPONENT is obtained by solving a similar optimization problem with the additional condition

$$\mathbf{w}_2'\mathbf{w}_1 = 0$$

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

$$\max_{\mathbf{w}_1} \{ \mathbf{w}_1' \mathbf{V} \mathbf{w}_1 : \mathbf{w}_1' \mathbf{w}_1 = 1 \}$$

Lagrangean function

$$L(\mathbf{w}_1, \lambda_1) = \mathbf{w}_1' \mathbf{V} \mathbf{w}_1 - \lambda_1 (\mathbf{w}_1' \mathbf{w}_1 - 1)$$

$$\frac{\partial L(\mathbf{w}_1, \lambda_1)}{\partial \mathbf{w}_1} = 2\mathbf{V}\mathbf{w}_1 - 2\lambda_1 \mathbf{w}_1 = \mathbf{0},$$

$$\frac{\partial L(\mathbf{w}_1, \lambda_1)}{\partial \lambda_1} = 1 - \mathbf{w}_1' \mathbf{w}_1 = 0,$$

$$(\mathbf{V} - \lambda_1 \mathbf{I}) \mathbf{w}_1 = \mathbf{0},$$

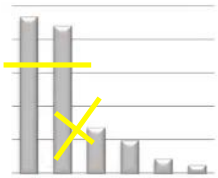
$$\mathbf{w}_1' \mathbf{V} \mathbf{w}_1 = \lambda_1 \mathbf{w}_1' \mathbf{w}_1 = \lambda_1 \mathbf{u}_1' \mathbf{u}_1 = \lambda_1$$

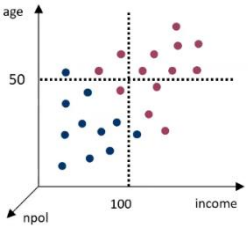
$$\mathbf{p}_j = \mathbf{X}\mathbf{u}_j$$

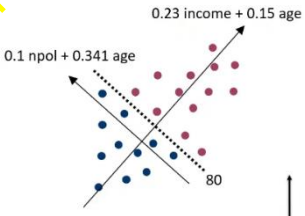
percentage of explained variance

$$I_q = \frac{\lambda_1 + \lambda_2 + \cdots + \lambda_q}{\lambda_1 + \lambda_2 + \cdots + \lambda_n}$$

scree-plot

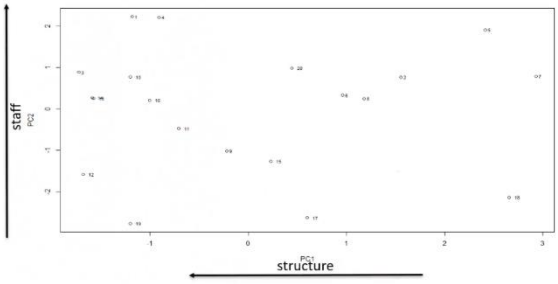


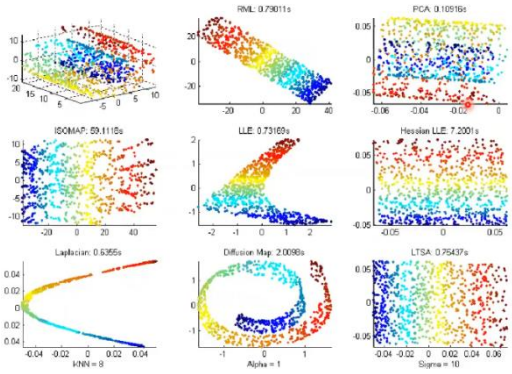


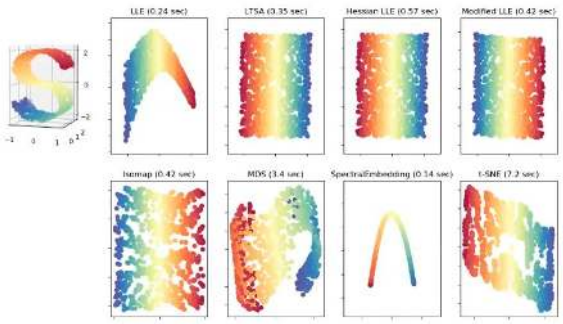


PCA can help the algorithm to have more accuracy, but you lose in interpretability (and usually accuracy is more important).

Sometimes you gain in interpretability if you find a way to summarize each principal component (similar to factor analysis). Useful for peer evaluation.







Nonlinear dimensionality reduction. You want to place the 3D dataset into a two-dimensional space preserving the order and "closeness" of different classes. The one that worked better is isomap because it uses all the plane to place the points.

1.3.2.3 Discretization or Aggregation

1.3.2.3.1 Discretization

Discretize numerical variables to reduce the number of unique values. You can choose class of equal size 😊, equal width 😐 , or a subjective discretization ☹.

1.3.2.3.2 Aggregation

For categorical variables, also called hierarchical discretization, it moves up into the category (for example geographically aggregating regions into a country).

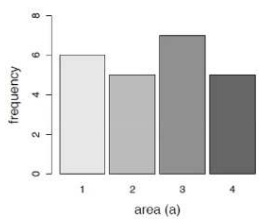
Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

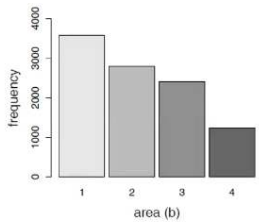
2. Exploratory Data Analysis

2.1 Uni-Variate Analysis

2.1.1 Categorical Attributes

2.1.1.1 Graphical Analysis





empirical frequencies

$$f_h = \frac{e_h}{m} \quad h \in \mathcal{H}$$

for large samples

$$f_h \approx \Pr\{x = v_h\} \quad h \in \mathcal{H}$$

Vertical bar analysis.

Category on the X axis and frequency on the Y. For small samples it can assume irregular shapes. For large samples it is a good approximation of probability of assuming a specific value.

2.1.1.2 Heterogeneity Index

GINI INDEX

$$G = 1 - \sum_{h=1}^{H} f_h^2$$

ENTROPY INDEX

$$E = -\sum_{h=1}^{H} f_h \log_2 f_h$$

heterogeneity max

$(H-1)/H$

0

heterogeneity min

Max heterogeneity → All values unique.

Min heterogeneity → One single unique value.

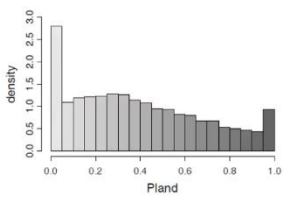
Binary attribute → H=2, G ranges from 0 to ½.

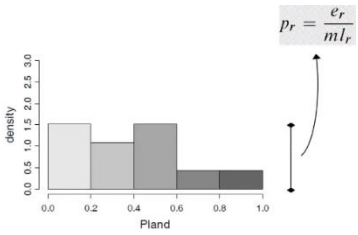
Binary attribute → H=2, E ranges from 0 to 1.

There will be a third index: misclassification index

2.1.2 Numerical Attributes

2.1.2.1 Graphical Analysis





$$p_r = \frac{e_r}{m l_r}$$

The shape depends strongly on the bins size and on the number of observations in the sample.

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

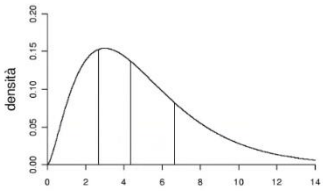
2.1.2.2 Measures of Central Tendency

SAMPLE ARITHMETIC MEAN:

$$\bar{\mu} = \frac{x_1 + x_2 + \cdots + x_m}{m} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

MEDIAN: central value of the observations

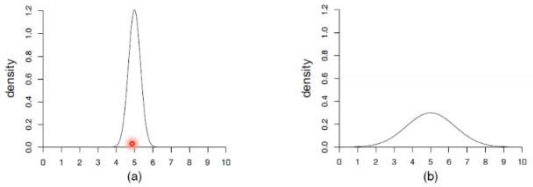
MODE: value that corresponds to the peak of the empirical density curve



Mean is too sensitive to outlier. The median is more robust. The mode is not well defined for bimodal distributions.

2.1.2.3 Measures of Dispersion

Measures of central tendency cannot spot the difference between these two distributions. That is why we need measures of dispersion. The range is a measure of dispersion but is not very accurate neither. The best one are variance and standard deviation or even better the coefficient of variation since it is an absolute value.



RANGE:

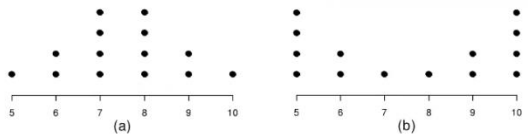
$$x^{range} = x^{max} - x^{min} \qquad x^{max} = \max_i x_i \qquad x^{min} = \min_i x_i$$

MEAN ABSOLUTE DEVIATION:

$$\text{MAD} = \frac{1}{m}\sum_{i=1}^{m} |x_i - \bar{\mu}|$$

SAMPLE STANDARD DEVIATION:

$$\bar{\sigma} = \sqrt{\bar{\sigma}^2}$$



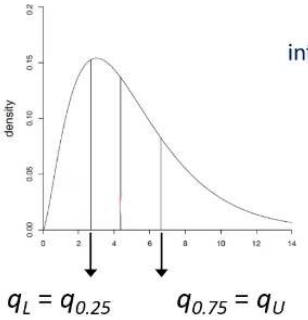
SAMPLE VARIANCE:

$$\bar{\sigma}^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \bar{\mu})^2$$

COEFFICIENT OF VARIATION:

$$\text{CV} = 100\frac{\bar{\sigma}}{\bar{\mu}}$$

2.1.2.4 Measures of Relative Location



interquartile distance

$D_q = q_U - q_L$

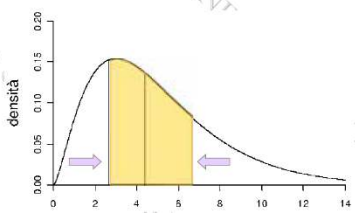
They provide even more detail on the dispersion of the attribute. They leave 25% of observation to the left or to the right. They allow to create new measures of dispersion and of central tendency.

The interquartile distance is actually a measure of dispersion.

MEAD-MEAN: mean of the values between q_L e q_U

TRIMMED-MEAN: mean of the values between q_P e q_{1-p}

WINSORIZED-MEAN: increase (decrease) values less (greater) than the quantile q_P (q_{1-p})



Mead-Mean is a bit too extreme.

Trimmed-Mean is less extreme.

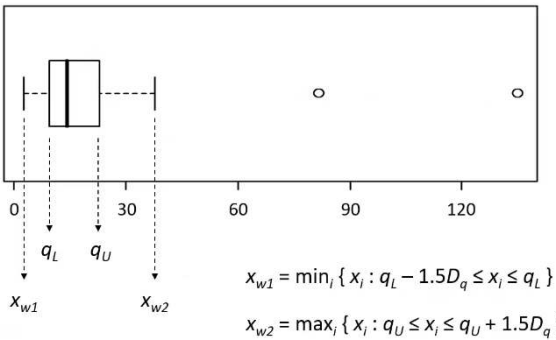
Winsorized-Mean works really well with extreme distribution.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

2.1.2.5 Analysis of the Empirical Density

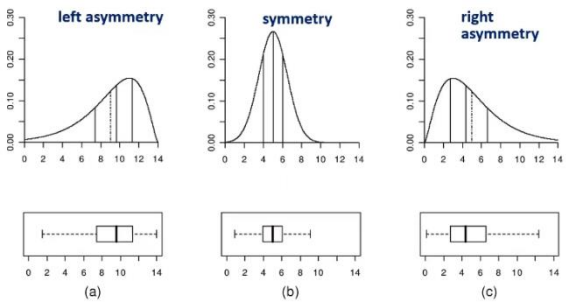
2.1.2.5.1 Box and Whiskers

This graphic tool provides much information: median, quartiles, symmetry and possible outliers.



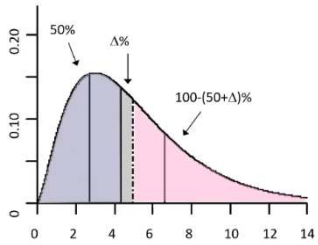
$$x_{w1} = \min_i \{ x_i : q_L - 1.5D_q \le x_i \le q_L \}$$

$$x_{w2} = \max_i \{ x_i : q_U \le x_i \le q_U + 1.5D_q \}$$



2.1.2.5.2 Asymmetry Index

ASYMMETRY index : $I_{as} = \frac{\bar{\mu}_3}{\bar{\sigma}^3}$ n $\quad \bar{\mu}_3 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{\mu})^3$



$I_{as} > 0$ right asymmetry

$I_{as} < 0$ left asymmetry

$I_{as} = 0$ symmetric

Usually it is better to use data vis (like box and whiskers) but if you want a numerical indicator there is the asymmetry index.

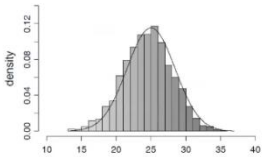
Asymmetry is also called skewness.

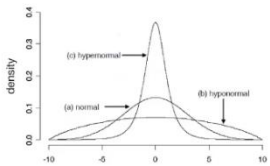
2.1.2.5.3 Normality

2.1.2.5.3.1 Kurtis Index

KURTOSIS

$$I_{\text{curt}} = \frac{\bar{\mu}_4}{\bar{\sigma}^4} - 3 \qquad \bar{\mu}_4 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{\mu})^4.$$

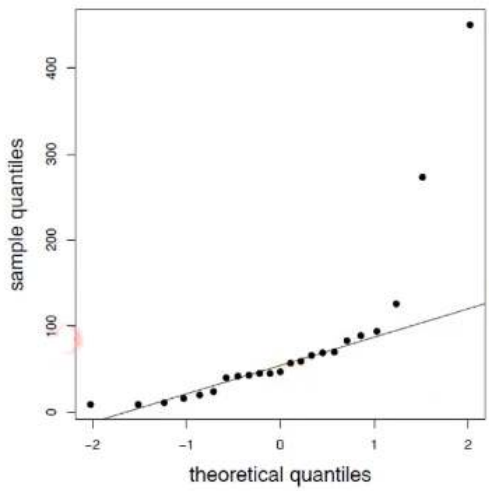




Depending on the signum of the Kurtis Index you can spot hyper-normality and hypo-normality.

2.1.2.5.3.2 Normal Probability Plot

Very powerful because it is a visual tool.



This document is available on studocu

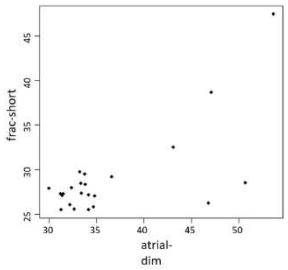
Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

2.2 Bi-Variate Analysis

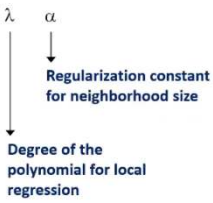
2.2.1 Graphical Analysis

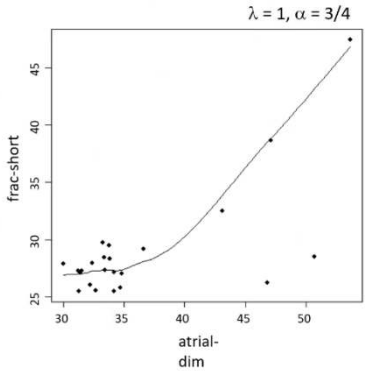
2.2.1.1 Scatter Plot

In this example you are tempted to draw conclusion very influenced by the presence of outliers.



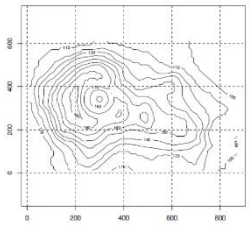
2.2.1.2 Loess Diagram





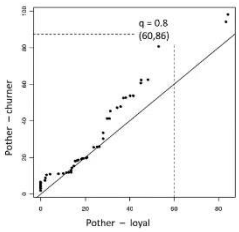
Changing the parameters we can refine our graph to better describe the behavior of the two attributes.

2.2.1.3 Contour Lines



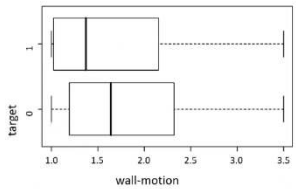
Planar representation of the relationship between 3 variables.

2.2.1.4 Quantile-Quantile Plot



Useful to make a comparison between two distributions. In particular, you can compare two subsets of the entire sample on the same attribute to check whether the two subsets behave in the same way.

2.2.1.5 Box and Whiskers



Also useful in variable selection phase before to feed it into the algorithm. If the two subsets have the same distribution you could discard that attribute.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

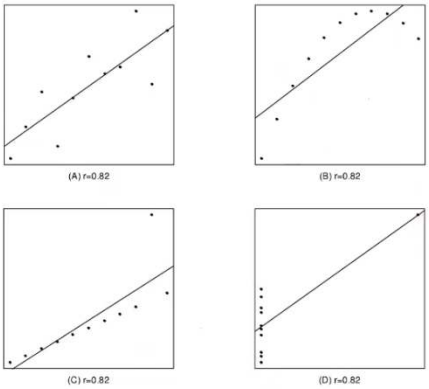
2.2.2 Correlation Measures

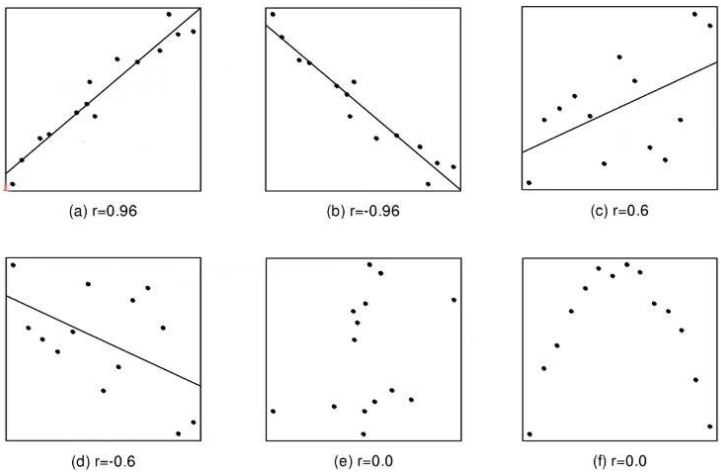
SAMPLE COVARIANCE:

$$v_{jk} = \mathrm{cov}(\mathbf{a}_j, \mathbf{a}_k) = \frac{1}{m-2}\sum_{i=1}^{m}(x_{ij} - \bar{\mu}_j)(x_{ik} - \bar{\mu}_k)$$

SAMPLE LINEAR CORRELATION :

$$r_{jk} = \mathrm{corr}(\mathbf{a}_j, \mathbf{a}_k) = \frac{v_{jk}}{\bar{\sigma}_j \bar{\sigma}_k} \qquad r_{jk} \in [-1, 1]$$





Checking linear correlation coefficient alone can be misleading.

2.2.2.1 Contingency Tables

	family		
area	0	1	totale
1	2	4	6 (f_1)
2	4	2	6
3	2	5	7
4	3	3	6
totale	11 (g_1)	14 (g_2)	25

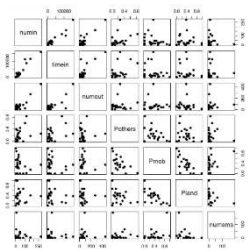
To check whether there is a relationship within two categorical attributes.

Two attributes are independent if

$$\frac{t_{r1}}{g_1} = \frac{t_{r2}}{g_2} \qquad r = 1, 2, \ldots, J$$

2.3 Multivariate Analysis

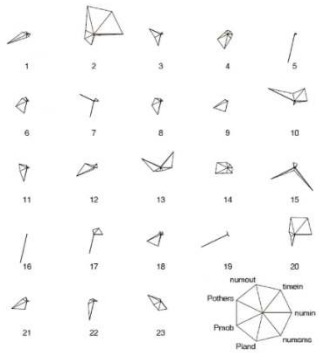
2.3.1 Scatter Plot



This document is available on studocu

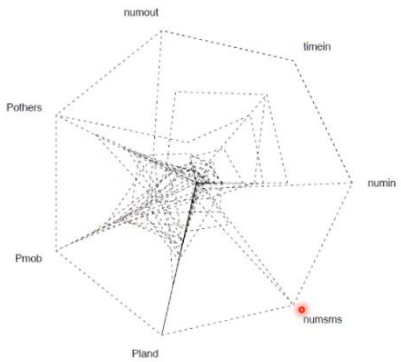
Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

2.3.2 Star Plot



Can be used only when very few observations. Only for numerical attributes.

2.3.3 Spider Web Charts



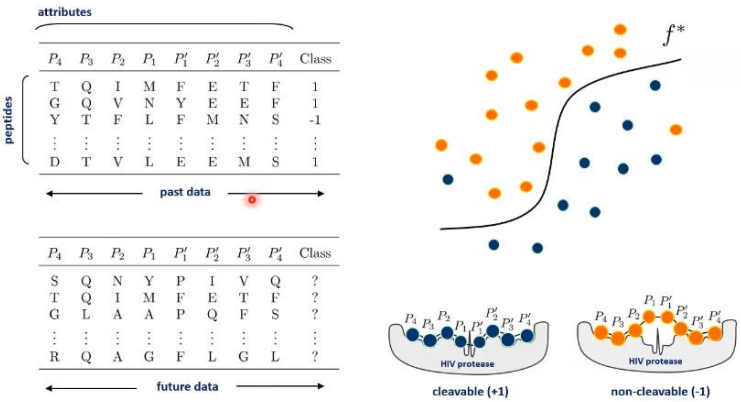
Particularly effective when there are many observations since you can really spot darker areas in most common values.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3. Classification

3.1 Key Concepts

Supervised learning problem in which the target variable is categorical.



Interesting example on HIV. There were 20 proteins in 8 positions, too many combinations to test all of them. Through a classification problem it was possible to predict whether untested combinations of proteins could stop the virus from reproducing or not.

3.1.1 Majority Vote Principle

How do we decide how to label one region once identified it? Following this principle, each new observation in that region will be labelled as the majority of observation in that region are labelled.

3.1.2 Classification as a Learning Task

Identification of recurring pattern in data for two goals: make prediction and improve understanding of the phenomenon of the domain experts.

3.1.2.1 Data Format

Data may take many different forms (vectors, images, text, genome ecc.)

3.1.2.2 Application Domain

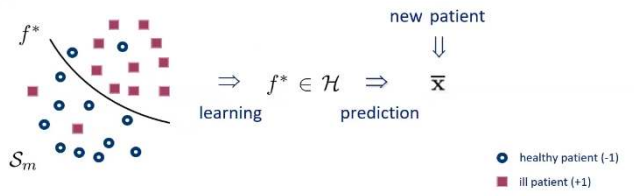
Many (Marketing, Security, Healthcare, SM analysis)

3.1.2.3 Mathematical Formulation

This document is available on studocu

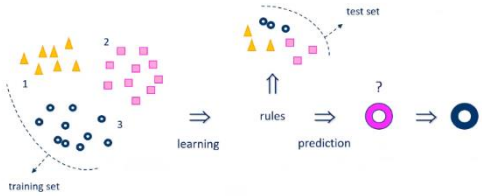
Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

- $\mathcal{S}_m = \{(\mathbf{x}_i, y_i),\ i \in \mathcal{M}\}$: training set, where $\mathbf{x}_i \in \Re^n$ and $y_i \in \mathcal{D}$
- $\mathcal{H}$ denotes a set of functions $f(\mathbf{x}) : \Re^n \mapsto \mathcal{D}$
- Classification problem: define a hypotheses space $\mathcal{H}$ and a function $f^* \in \mathcal{H}$ which optimally describes the relationship between $\mathbf{x}_i$ and y_i



All variable (also categorical ones) can be transformed into numerical ones applying dummy variables.

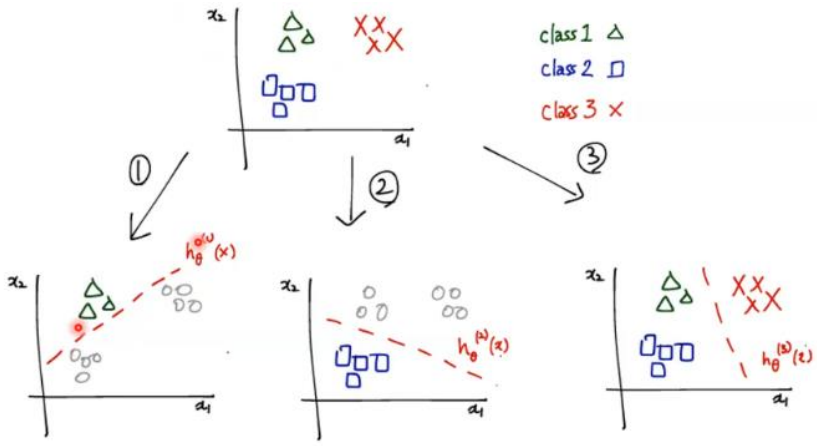
H is the hypothesis space, and it is only a subset of all possible functions, for example only polynomial up to grade 2.



3.1.3 Multi-Class Approaches

Not al problems are binary. But any multi-category problem can be reduced to a series of binary problems. There are two schemes to do so.

3.1.3.1 One Against All



We ask to each classifier to identify the region to separate one category from all the remaining ones. We have to solve K (number of categories of target variable) problems instead of just one, but they are binary problem so much easier than multi-class problems.

How do we perform final classification for new observation? We ask first classifier how to classify new observation, adding one score to the predicted category. We move to the next classifier and repeat the process adding new scores.

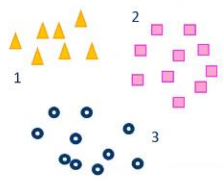
We than apply majority voting principle classifying the new observation to the higher scored category. We also can estimate the likelihood of the new observation to belong to each category (it is not a probability because not always obey to probability rules) dividing category score by total scores.

3.1.3.2 All Against All

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

Each classifier contrasts each category against one other. The problems become K(K-1)/2. Is it worth to pay the extra price? Usually yes but it depends on the problem.

Predictions are assigned through majority voting principle.



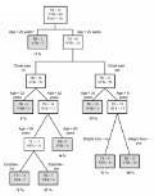
- Train K binary classifiers
  - Class 1 against {2,3,...K}
  - Class 2 against {1,3,...K}
  - .....
  - Class K against {1,2,...K-1}
- Predict for a new record by applying the K trained classifiers and assigning the label by majority voting

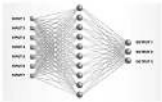
- Train K(K-1)/2 binary classifiers
  - Class 1 against {2}
  - Class 1 against {3}
  - .....
  - Class K-1 against {K}
- Predict for a new record by applying the K(K-1)/2 trained classifiers and assigning the label by majority voting

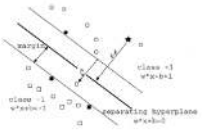
3.1.4 Taxonomy of Classification

Borders are not strict, but taxonomy is useful to understand.

- Heuristic methods
  - classification trees
  - nearest neighbor
- Separation methods
  - perceptron
  - neural networks
  - support vector machines
- Regression methods
  - logistic regression
- Probabilistic methods
  - bayesian methods







$$y_{MAP} = \arg\max_{y \in \mathcal{R}} P(y|\mathbf{x}) = \arg\max_{y \in \mathcal{R}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

Heuristics are based on common sense. There is no complex math, but in some situation, they work even better than more complex methods. For things as image recognition they will probably loose but for other problems work perfectly.

Separation can be linear and nonlinear.

Logistic regression is an adaptation of regression to classification problems.

3.1.5 Assessing Classification Methods

- Prediction accuracy
- Speed
  - time required to train a model
- Robustness
  - handling noise and missing data
- Scalability
  - effectiveness in handling large datasets
- Interpretability
  - value of the knowledge extracted from the model
- Rules effectiveness
  - number of rules
  - conciseness and significance of the rules

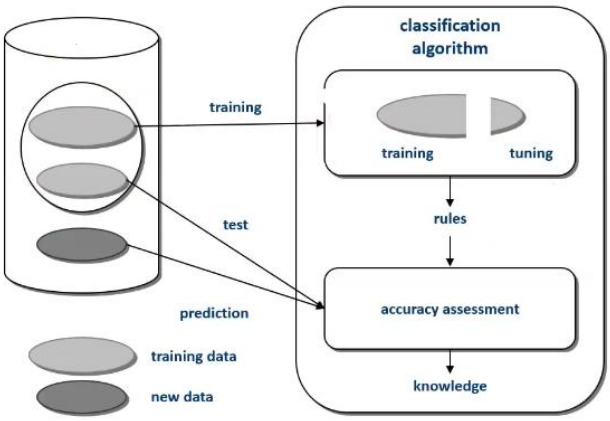
Scalability is strongly related with speed.

Rules means separation functions. They should be few functions, depending on few variables.

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.1.5.1 Prediction Accuracy



The tuning set is used only sometimes, and it is something internal to the algorithm, relevant for example to classification trees.

Any ML prediction can be assumed true only if the future is similar to the past, but this is an assumption that we have to take.

You can also update the accuracy when the future became the past and eventually re-train the algorithm.

3.1.5.1.1 Indicators

3.1.5.1.1.1 Loss Function

Loss function counting the number of misclassifications

$$L(y, f(\mathbf{x})) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}) \\ 1 & \text{if } y \neq f(\mathbf{x}) \end{cases}$$

3.1.5.1.1.2 Empirical Error

$$R_{\text{emp}} = \frac{1}{m}\sum_{i=1}^{m} L(f(\mathbf{x}_i))$$ **average number of miscalssifications**

3.1.5.1.2 Splitting Methods (Train and Test Sets) **Prediction Accuracy**

IMPORTANT: All of those techniques are useful just to get which model (intended as one between different classes of algorithm) better works with the data, estimating the accuracy many times and averaging it. Once you select one model, you re-train the selected model on all the data to make predictions.

3.1.5.1.2.1 HOld-Out Method **Hold Out**

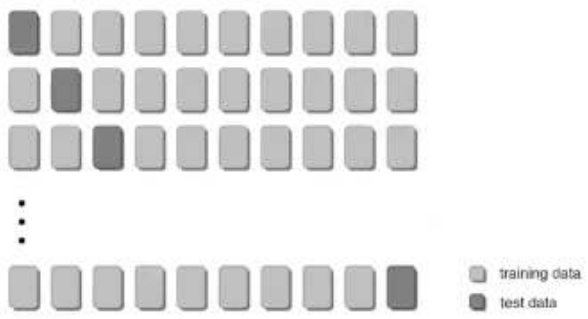
You decide a percentage (usually 25-30%) of observation to form the test set and then you random sample the whole set. The accuracy depends on the random observations sampled, so it is not very robust.

3.1.5.1.2.2 Repeated Random Sampling Method

Repeated application of hold one out method, the accuracy is then obtained by averaging al the accuracies.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.1.5.1.2.3 K-fold Cross Validation Method



K = 5 or 10

Each row is the whole dataset.

Each time you evaluate the accuracy with 9 portions as train and 1 as a test set. It depends even less from random sampling because it is a systematic procedure and you can be sure than any observations belongs 9 times to train and 1 to test.

3.1.5.1.2.4 Leave one Out Method

Limit of k-fold cross validation when k equals to the number of observations. It is the most accurate to assess the accuracy, but it is much heavier as computational time since there are many models to train and using many data (m-1).

Empirically it is better to use cross validation with small/medium size datasets (0 to 100k observations), with bigger datasets (millions of observation) you can just use hold out, since the test set being so large is unsensible to fluctuations.

3.1.5.1.3 Advanced Accuracy Indicators

Accuracy is not robust especially when there is an imbalance in the classes of the target attribute. The cost of misclassification are not constant, for example in disease identification is so much better to have false positive then to under estimate the illness of someone who might die.

3.1.5.1.3.1 Confusion Matrix

- p: true negative
- q: false positive
- u: false negative
- v: true positive

		predictions		
		−1 (negative)	+1 (positive)	total
examples	−1 (negative)	p	q	$p+q$
	+1 (positive)	u	v	$u+v$
	total	$p+u$	$q+v$	m

Accuracy = % records correctly classified= (p + v)/m

Recall (true positive rate tp) = % true positives correctly classified = v /(u + v)

Precision (prc) = % true positives among those predicted positives = v /(q + v)

False positive rate = q/(p+q)

True negative rate = p/(p+q)

False negative rate = u/(u+v)

F-measure $F = \frac{(\beta^2 + 1)\,\mathrm{tp} \times \mathrm{prc}}{\beta^2\,\mathrm{prc} + \mathrm{tp}}$.

Geometric mean $\mathrm{gm} = \sqrt{\mathrm{tp} \times \mathrm{prc}}$.

Usually the positive is the one we are interested in. Those measures are expressed on the test set.

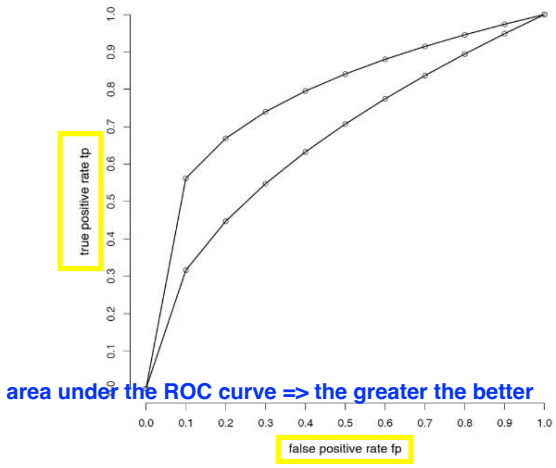
The F-measure is a harmonic mean between true positive rate and precision.

بتا رو ما فرض میگیریم. فکر کن وقتی هیچ سنسی نسبت به کاست نداریم میگیریم ۱
بتا رو به صورت رفلکت ریشوی بین کاستشون پیدا میکنیم. خیلی دقیق نگفت چطوری

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.1.5.1.3.2 ROC Curve **recieving operating carachtersitics**

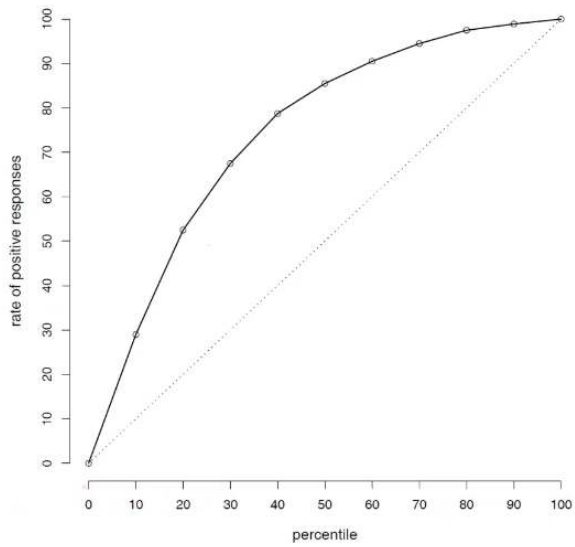


Every classifier associate to each prediction a score (the likelihood to be positive). You can decide the threshold, which is the score above which an observation is predicted positive (and is not always 0.5). In ROC curve you put TP (Recall) in vertical axis and FP (1-Precision) in horizontal axis. You plot the TP and FP for each threshold value from one to zero (of course a threshold closer to 0 give more importance to Recall while a threshold closer to 1 give more importance to Precision.

In other words you rank observation from higher score to lower score. You divide in deciles, the first decile will have a high TP and a low FP. Then they will both increase but at different peace (Concave Curve).

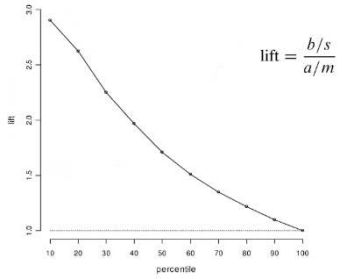
It is useful to compare different algorithm independently from the chosen threshold. You consider the AUC (the closer to 1 the better, 0.5 means pure randomness and close to 0 worse than random).

3.1.5.1.3.3 Cumulative Gain



On the x axis you rank the observation by descending score divided in percentiles. On the y axis the percentage of total positive observation present in that percentile. In the first decile you will find the highest percentage of positive, lowering the threshold the absolute number will increase but the slope will diminish (Concave Curve). For pure random selection, the curve follows the diagonal since on 10% of the observations randomly picked you would find 10% of the positive (lower limit for any meaningful algorithm).

3.1.5.1.3.4 Lift



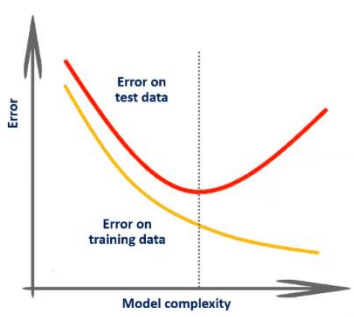
$$\text{lift} = \frac{b/s}{a/m}$$

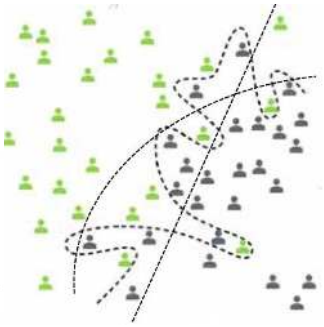
The lift curve indicates how much your algorithm differ from pure random selection.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

if accuracy of training set is larger than test set ==>>>> overfitting Alert

3.1.6 Overfitting





To avoid overfitting we limit the complexity of the hypothesis space from which we extract our separation function. We give up some accuracy to make our result more robust to new data (starting from the test set). When accuracy in train and test differs too much there is overfitting.

3.1.7 Numerical representation of Categorical variables

- variable X_j assumes values in the set $V = \{v_1, v_2, \ldots, v_H\}$
- H-1 dummy variables

$$D_{j1}, D_{j2}, \ldots, D_{j,H-1}$$

- For example, to represent the month associated to an observation

$$\{D_{gen}, D_{feb}, D_{mar}, D_{apr}, D_{mag}, D_{giu}, D_{lug}, D_{ago}, D_{set}, D_{ott}, D_{nov}\}.$$

You cannot just give a code to each category (1, 2, 3, …, H) because the algorithm would interpret it as an ordinal variable. So you need an orthonormal representation (dummy variable), which means a vector of binary attributes. The last one is dependent from all the others so it can be deleted. For classification, the only advantage is to have one less variable, and since now we have pretty powerful machine is better to keep them all. Another story for regression where two dependent variables are a big problem, and you need to drop one.

3.2 K-Nearest Neighbors

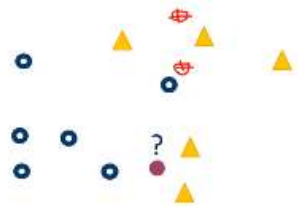
To a new record is attributed the class to which belong the majority of records out of the K closest ones (majority voting)

The choice of K affects model robustness and accuracy

By increasing K …

less sensitive to outliers

less distinction among classes



Very simple, does not even require training. To make a prediction you use the majority vote principle, selecting the class of the majority of the k nearest neighbors.

K is a hyper parameter, there is no theorem that states which values is optimal, you have to try with many values (3,5,7,9, …) and find the one that get better results on the test set. You choose odds values to avoid balance in counting votes. Intuitively, the score is the number of votes divided by k.

Hyper Parameters: K, Measure of distance.

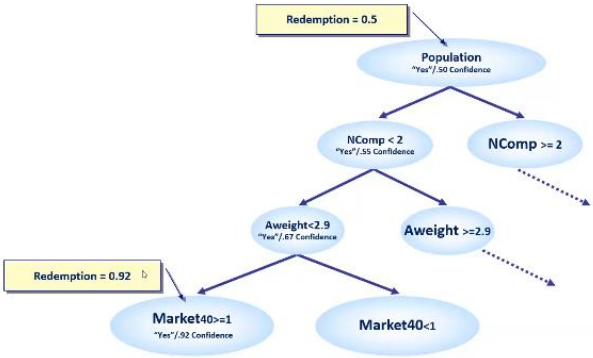
This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.3 Classification Tree

3.3.1 General Introduction

Is very similar to human way of reasoning.



The root node is the first one and all the observation are placed there, is the only one that is generated by the algorithm automatically. Leaves are the final node, where predictions are made by majority vote principle (and scores are assessed by the percentage of positive). In the root node there is the situation of maximum confusion. Using Gini or entropy index we can measure confusion and try to minimize it. How? Analyzing each attribute and each possible split the algorithm select the attribute and the splitting threshold that minimize the choose index. The possible splitting values are not an infinite number since it makes sense to try a value only if there is at least one observation at each side of that value, so the maximum number of trials is m (number of observation) for the split value and n (number of attributes) for the attribute. The algorithm is linearly dependent from m and n, making it pretty fast.

3.3.1.1 Characteristics of Classification Trees

- Base recursive greedy algorithm
  - All records are placed in the root node
  - Records in each node are split according to a heuristic test based on the value of one or more attributes

not optimal globali

Recursive: at each node perform exactly the same operation.

Greedy: choose each time what it appears to be the best choice, not sure it is the actual optimal one.

- To prevent overfitting
  - pre pruning by stopping criteria
    - purity threshold
    - minimum cardinality
    - minimum information gain

evaluate the accuracy on test set. the accuracy without the last brancing was same as before, may decide to post pruning. it's better on test so we are doing overfitting.

  - post pruning: identifies and removes ramifications affected by noise

hyper params:
- purity treshhold (90%)
- minumim cardinality - minimum number in a leaf to split
- information gain - difference of hetrogenieity reduction.

3.3.1.2 Pre-Pruning (Stopping Rules)

Pruning means to cut a branch after a node, so that it becomes a leaf.

3.3.1.2.1 Purity threshold

Purity Threshold limits the percentage of observation of the same target class in one leaf (for example 90% of positive in one node will cut that branch).

3.3.1.2.2 Minimum Cardinality

Minimum number of observations in one node. If the split proposed by the algorithm lead to a node populated by less observation than the limit, the split does not happen.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.3.1.2.3 Minimum Information Gain

If the index does not improve enough from one node to the other, you do not perform the split.

3.3.1.3 Post Pruning

The train-set is divided internally and a portion is left apart for internal tuning, self-testing and self-assessing its performance. The algorithm performs an internal prediction with the tuning data, checking whether the accuracy is higher with or without the last split, and in case it removes it checking next the split above. It is pretty heavy as line of code of the algorithm and of time for running the algorithm.

3.3.1.4 Taxonomy of Classification Trees

- **Classification trees are distinguished in:**
  - **binary trees**
  - **general trees**

  and also in

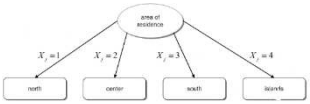
  - **univariate trees (or axis parallel)**

$$\Rightarrow \quad X_j \leq b$$

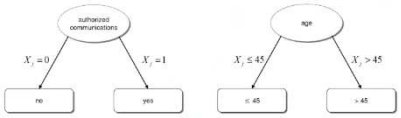
  - **multivariate trees (or oblique)**

$$\Rightarrow \quad \sum_{j=1}^{n} w_j x_j \leq b$$

3.3.1.4.1 Binary Vs General Trees

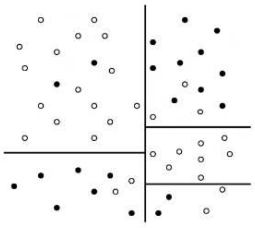


General Trees.



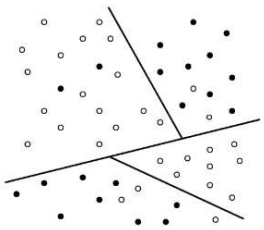
Binary Trees. (More Common ones)

3.3.1.4.2 Univariate Vs Multivariate Trees



classification by an axis parallel tree

better interpretability



classification by an oblique tree

In univariate trees the splitting rule is based on a single attribute, while in multivariate can be a linear combination of many of them. Axis parallel usually have worst performance but more interpretability.

Usually we give more importance to accuracy.

3.3.2 Mathematical Formulation (axis parallel only)

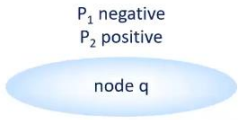
In axis parallel trees the best separation identifies the most effective attribute and the correspondent split to minimize the index, based on a greedy logic.

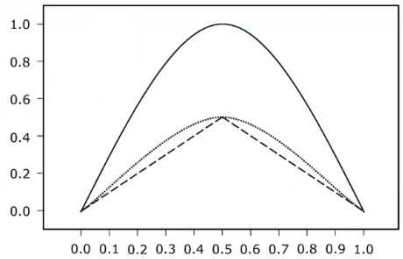
This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.3.2.1 Measures of heterogeneity

- **misclassification index** $\text{Miscl}(q) = 1 - \max_h p_h$
- **entropy index** $\text{Entropy}(q) = -\sum_{h=1}^{H} p_h \log_2 p_h$
- **Gini index** $\text{Gini}(q) = 1 - \sum_{h=1}^{H} p_h^2$





X axis: Probability of being positive/negative

Y axis: Index

From top to bottom: Entropy, Gini, Misclassification

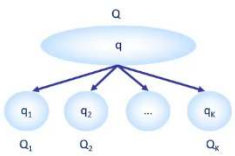
3.3.2.2 Splitting Rule

- **Impurity of a splitting rule**

$$I(q_1, q_2, \ldots, q_K) = \sum_{k=1}^{K} \frac{Q_k}{Q} I(q_k).$$

- **At each node select the rule minimizing the impurity or, equivalently, maximizing the information gain**

$$\Delta(q, q_1, q_2, \ldots, q_K) = I(q) - I(q_1, q_2, \ldots, q_K)$$
$$= I(q) - \sum_{k=1}^{K} \frac{Q_k}{Q} I(q_k).$$



Q = Num. of observation at node q

Q1 = Num. of observation at node q1

3.3.2.2.1 Impurity Index:

I = Heterogeneity Index (Entropy, Gini, ecc…)

The impurity index is a weighted average of I based on the number of observations at each node qk.

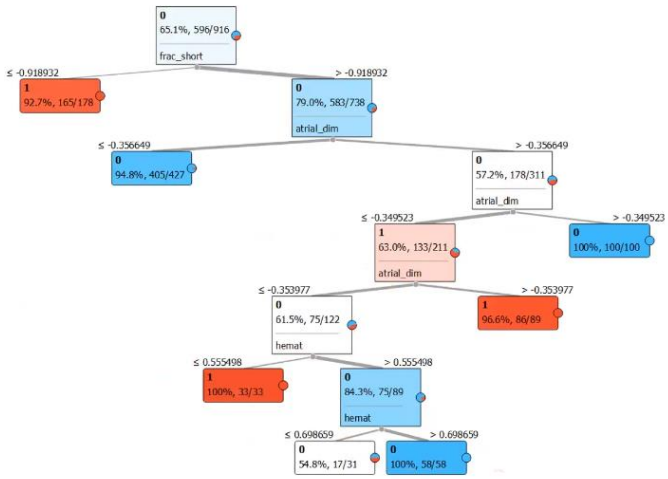
The algorithm performs the split that minimize the impurity index.

3.3.2.2.2 Information Gain

Alternatively, it performs the one that maximise information gain.

3.3.3 Hyper parameters

Hyper parameters: Number of ramifications, Purity threshold, Minimum cardinality, Minimum information gain, Heterogeneity index.



Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.4 Bayesian Methods

- **compute posterior probability:**

$$P(y|\mathbf{x})$$

- **estimate of prior probability**

$$P(y) = P(y = v_h) = \frac{m_h}{m}$$

- **compute posterior using Bayes theorem**

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{\sum_{l=1}^{H} P(\mathbf{x}|y)P(y)} = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

Not heuristic method, but based on probabilistic theory (Parametric). It tries to estimate the posterior probability, which is the probability of having an outcome (positive or negative) given the vector of explanatory variables X.

Example. Bayes theorem can be very useful in identifying the effectiveness of a medical test of identifying positive case of COVID-19. Y is the positive outcome of the test given X the real state of the patient (positive). Using Bayes theorem, you can easily estimate the possibility of X given Y doing many tests in laboratory: out of the patient who resulted positive (Y) how many were actually infected by COVID-19 (X)?

- **To classify new observations use *maximum a posteriori hypothesis* (MAP) principle:**

$$y_{MAP} = \arg\max_{y \in \mathcal{H}} P(y|\mathbf{x}) = \arg\max_{y \in \mathcal{H}} \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

To classify new observations you could just apply MAP and label them as the target category that maximize the numerator.

- **Maximize numerator**

$$P(\mathbf{x}|y = v_h)P(y = v_h) \geq P(\mathbf{x}|y = v_l)P(y = v_l), \quad l = 1, 2, \ldots, H, \; l \neq h$$

...but sample estimation of $P(\mathbf{x}|y)$ **cannot be done in practice**

The problem is that in real life you cannot estimate P of X given Y like you did for the medical test since X is a vector of very high dimension. So the probability of having a vector identical to the new one will likely be zero or very low.

3.4.1 Naïve Bayesian Methods

- **assume attributes are independent:**

$$P(\mathbf{x}|y) = P(x_1|y) \times P(x_2|y) \times \cdots \times P(x_n|y) = \prod_{j=1}^{n} P(x_j|y)$$

- **two possible cases:**
  - **categorical attributes:**

$$P(x_j|y) = P(x_j = r_{jk}|y = v_h) = \frac{s_{jhk}}{m_h}$$

  - **numerical attributes: postulate a given distribution, such as**

$$P(x_j|y = v_h) = \frac{1}{\sqrt{2\pi\sigma_{jh}}} e^{-\frac{(x_j - \mu_{jh})^2}{2\sigma_{jh}^2}}$$

Under the assumption of independence between attributes, the probability can be estimated as the product of the probability of each component. For categorical attributes, it is simply the relative frequency.

For numerical one you have to make an assumption on the distribution, assuming normal distribution the formula is the following. That's why Bayesian methods work better for categorical variables.

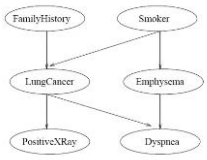
When doing a prediction, you just label the new observation with the one with the highest probability (and you consequently have all the scores summing up all the probabilities and the relative weight is the score). Those are not properly probability since they do not sum up to one because we did not consider the denominator (P of X).

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.4.2 Bayesian Belief Networks

BAYESIAN BELIEF NETWORKS allow to define dependency relationships between (small) subsets of attributes



Acyclic oriented graph

Conditional probability table

	FH,S	FH,~S	~FH,S	~FH, ~S
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

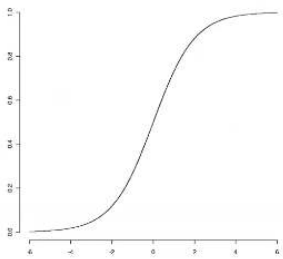
An effort to relax the assumption of independence between attributes. Particularly used in medical field since physicians are willing to introduce their personal knowledge of the topic (domain expert).

You can try to introduce the combined probabilities into the analysis to perform a more refined search.

3.5 Logistic Regression

- **A logistic function (sigmoid)**

$$S(t) = \frac{1}{1+e^{-t}}$$



Postulate posterior probability

$$P(y=0|\mathbf{x}) = \frac{1}{1+e^{\mathbf{w}'\mathbf{x}}}$$

$$P(y=1|\mathbf{x}) = \frac{e^{\mathbf{w}'\mathbf{x}}}{1+e^{\mathbf{w}'\mathbf{x}}}$$

Parametric approach (means that you have to make assumptions).

In this case the parametric assumption Is that the posterior probability (being negative given X) follows a sigmoid function. The second probability is a consequence of the first one since they must add up to one.

Odds ratio is the probability of being positive divided by probability of being negative. Taking the log of the odds ratio you obtain a linear regression model. Not very powerful when X is multi-dimensional.

January 1986: Explosion of the Space Shuttle Challenger

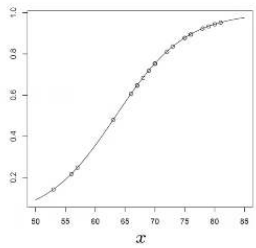
Launch temperature and *O Rings* failures

launch	temp.	failure	launch	temp.	failure	launch	temp.	failure
1	53	Y	9	68	N	17	75	N
2	56	Y	10	69	N	18	75	Y
3	57	Y	11	70	N	19	76	N
4	63	N	12	70	Y	20	76	N
5	66	N	13	70	Y	21	78	N
6	67	N	14	70	Y	22	79	N
7	67	N	15	72	N	23	80	N
8	67	N	16	73	N	24	81	N

Parameters estimate:

$\beta_1 = 0.17$

$\beta_0 = -10.8$ (intercept)



Odds Ratio = $\ln \frac{p(x)}{1-p(x)} = \beta_0 + \beta_1 x$

$p(x)$ **is the success probability (failure = N) when temp. =** x

Temperature when Challenger exploded was 31°F...

... the Odds Ratio was ?

$x = 31$ Odds Ratio = 0.0038 !!!

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.6 Support Vector Machines

Choose the function $f^* \in \mathcal{H}$ (hypothesis) to minimize the empirical risk

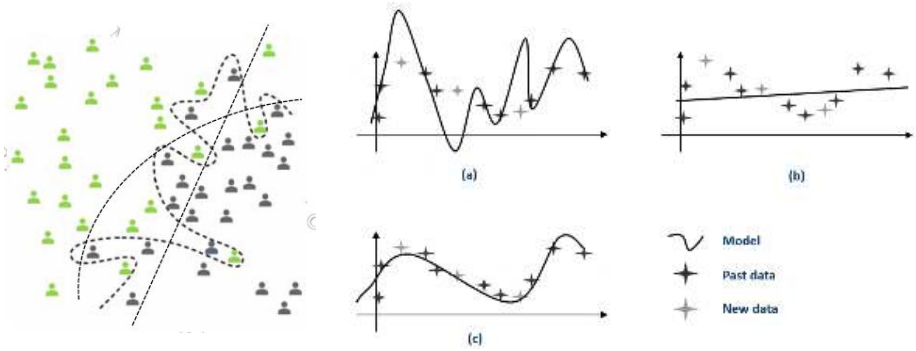
$$R_{emp}(f) = \frac{1}{m}\sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i))$$

Two critical issues:

- overfitting
- ill-posed optimization problem

The optimization problem to minimize empirical error (number of misclassifications) is ill-posed: small changes in the input data imply big changes in the solution, it is not robust.

3.6.1 Accuracy Vs. Generalization



3.6.2 Classification and Statistical Learning Theory

Choose the function $f^* \in \mathcal{H}$ (hypothesis) to minimize the empirical risk

$$R_{emp}(f) = \frac{1}{m}\sum_{i=1}^{m} V(y_i, f(\mathbf{x}_i))$$

Two critical issues:

- overfitting
- ill-posed optimization problem

Choose $f^* \in \mathcal{H}$ to minimize the structural risk (SRM principle)

$$\hat{R}(f) = \frac{1}{m}\sum_{i=1}^{m} \underbrace{V(y_i, f(\mathbf{x}_i))}_{\text{loss function}} + \underbrace{\lambda\|f\|_K}_{\text{regularizer}}$$

The solution is to minimize not the empirical risk but the structural risk, which is the empirical risk plus a regularize term which solve both problems (overfitting and ill-posed) since it takes care of the generalization capability which is related with the complexity of the hypothesis functional space.

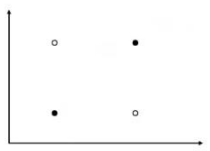
The regularization term is the norm of degree k of the function f. We have not studied that norm because it is functional analysis, in a functional space.

If f is a linear function the norm of f is just the norm of its coefficients.

Probabilistic upper bound on the expected risk (structural risk)

$$R(f) \le \sqrt{\frac{\gamma \log(2t/\gamma) + 1 - \log(\eta/4)}{t}} + R_{emp}(f)$$
$$\le \Psi(t, \gamma, \eta) + R_{emp}(f),$$

γ is the VC-dimension (Vapnik-Chervonenkis) expressing the complexity of the hypothesis space $\mathcal{F}$



- the VC dimension of the lines in the plane is 3
- the VC dimension of the hyperplanes in the n-dimensional space is $n+1$

The theoretical structural risk (the real one, as expected value, not the estimated one on the finite training set) cannot be calculated, but is limited from above by a probabilistic upper limit.

Probabilistic means that there is a 95%* probability (*Value depending on ETA, like a confidence interval depends on alpha), that it is lower than the bound.

The lower bound is the empirical risk while the upper bound is given by the empirical risk plus a function PSI, a function of t (number of observations in the training set), GAMMA (the complexity of the hypothesis space) and ETA.

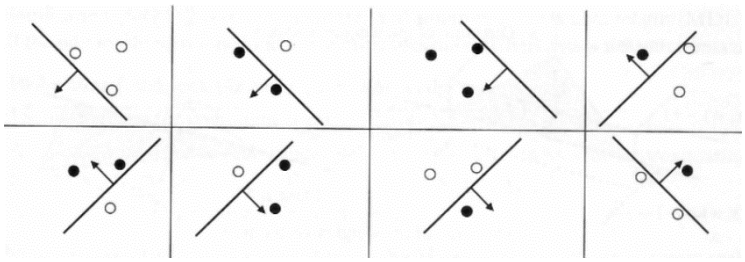
This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

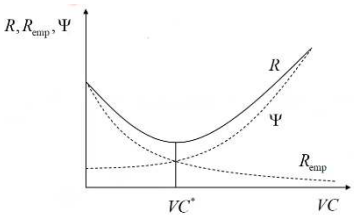
When t is large, PSI goes to 0 (a large training dataset the stricter is the approximation provided by the bound).

ETA: the higher the confidence in the probabilistic result, the higher goes the bound.

GAMMA is the complexity of the hypothesis space, when complexity increase, the more we risk overfitting, the higher goes the bound.



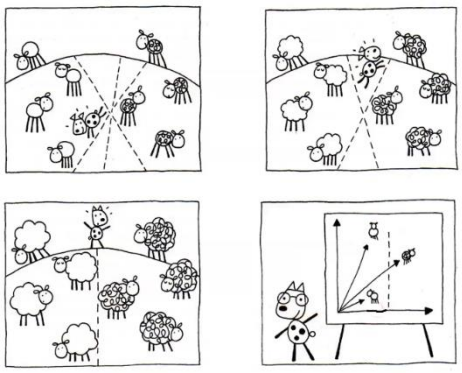
If a single function of our hypothesis functional space can properly separate N point (in a particular position that you can choose) between labelled positive and negative , for any possible combination of labels (2^N combinations), then that space has VC-dimension N.



The theoretical structural risk is bounded below by empirical risk and PSI, but one increase and the other decrease when VC grows, so you have to find the best balance between the two choosing an appropriate hypothesis space, finding the VC that leads to the optimal (minimum) risk.

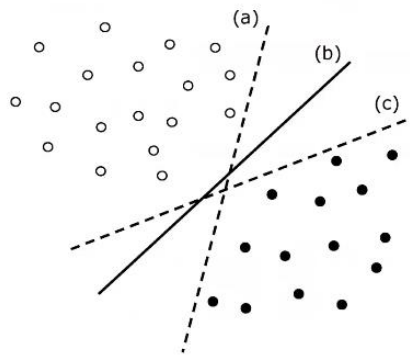
3.6.3 SVM Formulation

3.6.3.1 Dog and Sheep Example



Even when there are many linear trajectories that can correctly classify all the sheep, the middle one is better because is more robust, the other two are ill-posed (small change in sheep positions leads to misclassification). The best one is the one that has maximum minimum distance within the trajectory and the closest sheep. The minimum distance is margin of separation and we want to maximize this margin.

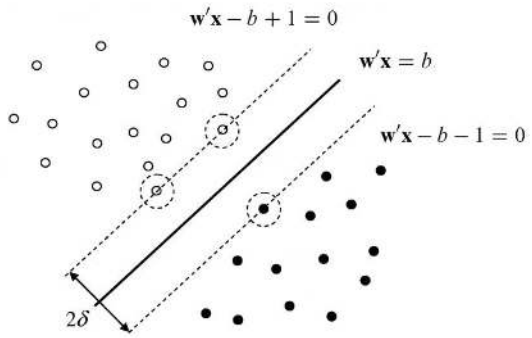
In the last example, the dog understands that there are three points more important than the others, the ones that has minimum margin (the ones that oblige the trajectory to pass exactly from there): those are the support vector. To find the best trajectory is a minimization problem.



Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.6.3.2 Linear Separation

3.6.3.2.1 Geometrical Interpretation



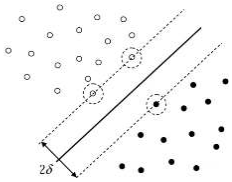
$$\delta = \frac{2}{\|w\|}, \qquad \|w\| = \sqrt{\sum_{j \in \mathcal{N}} w_j^2}$$

We know X, we want to find W and b.

The canonical hyperplanes are parallel to the separating hyperplane and they pass exactly on the closest points (1 can be always obtained dividing the whole equations by the original value). The margin of separation is DELTA, and it is equals to 2 over the norm of W (by simple analytical proprieties of lines and points, the steepest the line the closest are the canonical hyperplanes since the vertical distance is always one). We want to maximize DELTA, and we ca do it minimizing W.

3.6.3.2.2 Hard Separation Problem

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 \quad i \in \mathcal{M}$$



We get rid of the square root and we got our optimization problem, minimizing the square of the norm. Without constrain we would obtain a vector 0, which doesn't perform any separation: we need to force the points to lay on the two opposite sides of the separating line based on their label (+ or -).

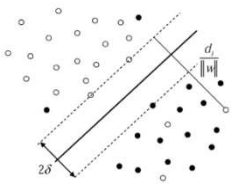
We need a constrain: the positive on one side of one canonical hyperplane and the negative on the other side of the other canonical hyperplane (not just the separating one, there must be the maximum empty space between the two*). The notation is -1 for negative and +1 for positive (we need this assumption also for artificial neural networks). Example for positive ones (the black): the constrain must be true for every point (i belongs to m); since the yi is +1 we obtain WXi-b-1 greater or equal to 0. This equation identifies all the region of space below the lower canonical line (for example equal to +3, if you take the 3 on the other side you get WXi-b-4=0). Same for the negatives.

It is a quadratic optimization problem with linear constrains we can easily solve it by direct methods, a variation of the simplex method which uses a projection of the gradient, in a very efficient way. Another approach is based on Lagrangean relaxation, we will use the latter because it works also with non-linear separation rules.

3.6.3.2.3 Relaxing the Problem for Outliers

What if we have an outlier? A point which is on the "wrong" side of the canonical hyperplane? At this stage, the optimization problem will have no feasible solution. We need to relax a bit the problem to also accept some misclassification.

$$\min_{\mathbf{w},b,d} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + \lambda \sum_{i=1}^{m} d_i$$
$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i - b) \geq 1 - d_i \quad i \in \mathcal{M}$$
$$d_i \geq 0 \quad i \in \mathcal{M}$$



We introduce a slack variable: d is the distance between the outlier and the correct canonical hyperplane (*It means that a point is considered an outlier and pay a penalty also when it falls into the "empty region"). When d is greater than zero, it allows the point to lay on the "wrong" side (but you have a penalty in the minimization (because of its positive signum).

LAMBDA is relative importance of d (misclassification) over W (generalization) → HYPERPARAMETER

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.6.3.2.4 How to solve the optimization problem: Lagrangean Relaxation and Duality

(useful to know in order to generalize to non-linear separation later)

$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_{i=1}^{m} d_i - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w}'\mathbf{x}_i - b) - 1 - d_i] - \sum_{i=1}^{m} \mu_i d_i$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = \mathbf{0},$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu})}{\partial b} = \lambda - \alpha_i - \mu_i = 0,$$

$$\frac{\partial L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu})}{\partial \mathbf{d}} = \sum_{i=1}^{m} \alpha_i y_i = 0,$$

Instead of considering the constrains, you put them into the objective function by multiplying them by the appropriate Lagrangean multiplier, an extra cost to pay if you violate a constraint. Then you will solve the dual problem.

The first constrain is multiplied by Lagrangean multiplier ALPHA and the second one by MI (there is one ALPHA and one MI for each observation, they are m-dimensional).

Since now is an unconstrained problem, to solve it just calculate partial derivatives and nullify them.

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i,$$

$$\lambda = \alpha_i + \mu_i,$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0.$$

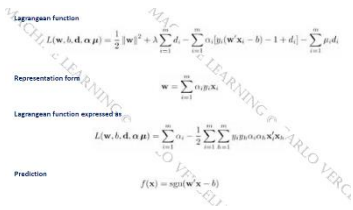
You obtain 3 equations (we are not really interested into the Lagrangean multipliers yet), the first one is called representation theorem: it tells us that the coefficients of the separating hyperplane W are a linear combination of the past data (X), the labels (Y) and of the dual variable ALPHA. If we know the ALPHAs, the problem is solved.

$$L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\, \boldsymbol{\mu}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{k=1}^{m} y_i y_k \alpha_i \alpha_k \mathbf{x}_i' \mathbf{x}_k$$

Thanks to the second and the third equations we can re write the Lagrangean function as a quadratic function in ALPHA.

This function is the objective function of the dual problem, so it must be maximized with respect to the Lagrangean multiplier. Very easy, because it is a quadratic in ALPHA with negative signum (parabolic function pointing downward, it makes sense to maximize and have a unique solution).

(Observation useful for later: the Lagrangean depends on the past data only in the form of inner product (Xi'Xh), and this will be used in kernels explanation).

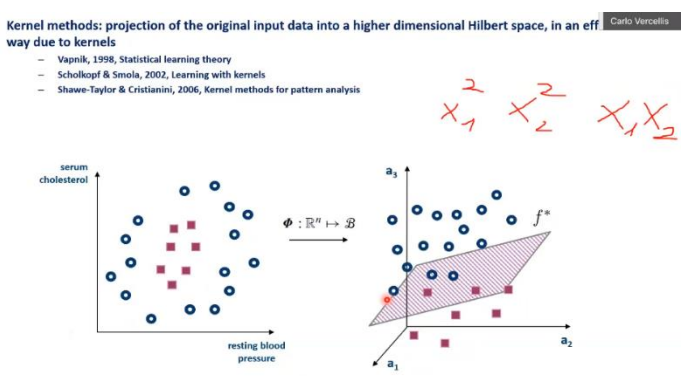


3.6.3.2.5 Predictions

We know the W, calculating the ALPHAs and replacing them into the representation theorem. When a new observation comes into play, you just calculate W'X-b, and the signum will simply tell you on which side of the separating hyperplane is the new point.

3.6.3.3 Non-Linear Separation with SVM

3.6.3.3.1 Main Idea



Project the original data, into a higher dimensional Hilbert space (infinite dimensional space). An example is a two-dimensional space, X1 and X2, you project the data into X1, X1^2, X2, X2^2, X1X2 (5 dimension), here any linear combination (hyperplane), when projected into the original space, is a quadratic form. Thanks to kernels, you can choose the appropriate and efficient transformation in order to perform an optimal separation using only linear function in the projected space.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

Same principle of Laplace and other transformations, we move to a different space to easily solve a problem and then go back to original space.

This time, thanks to kernels, we can perform this operation implicitly, is saying in the original space.

3.6.3.3.2 Kernels

Any function in the original n-dimensional space that, applied to two points in the original space, it is equivalent to project the two points in a new m-dimensional space (m>>n) and perform the inner product of the projected points. This is possible because the inner product results simply in a one-dimensional (scalar) number, so you do not really need to perform the transformation if you find the function that result in the same scalar number. So each space, each transformation, has its kernel function. (More Reading: https://towardsdatascience.com/kernel-function-6f1d2be6091)

Kernel $\boldsymbol{\Phi}(\mathbf{x}_i)'\boldsymbol{\Phi}(\mathbf{x}_h) = k(\mathbf{x}_i, \mathbf{x}_h)$

Lagrangean function $L(\mathbf{w}, b, \mathbf{d}, \boldsymbol{\alpha}\,\boldsymbol{\mu}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{h=1}^{m} y_i y_h \alpha_i \alpha_h k(\mathbf{x}_i, \mathbf{x}_h)$

- **Existence of meaningful kernel functions is guaranteed by Mercer Theorem**
  - **polynomial kernels** $k(\mathbf{x}_i, \mathbf{x}_h) = (\mathbf{x}_i'\mathbf{x}_h + 1)^d$
  - **radial basis kernels** $k(\mathbf{x}_i, \mathbf{x}_h) = \exp\left(\frac{-||\mathbf{x}_i - \mathbf{x}_h||^2}{2\sigma^2}\right)$
  - **neural networks kernels** $k(\mathbf{x}_i, \mathbf{x}_h) = \tanh(\kappa\mathbf{x}_i'\mathbf{x}_h - \delta)$

This is useful because the solution of the Lagrangean is dependent only on the inner product of ALPHA. If you repeat all the calculation above in the transformed m-dimensional space, you get to the same formula where there is the inner product of projected Xi and Xh, which is just the kernel function applied to Xi and Xh.

There are some typical Kernels for some typical spaces: polynomial to a space of degree d, radial basis (Gaussian) to an infinite-dimensional space, ecc… There are kernels specific that works better for each application field.

3.6.3.3.3 Predictions

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i \boldsymbol{\Phi}(\mathbf{x})'\boldsymbol{\Phi}(\mathbf{x}_i) - b\right)$$

Now we do not have a separation hyperplane but a separation surface. You easily substitute the product with the kernel of X, Xi.

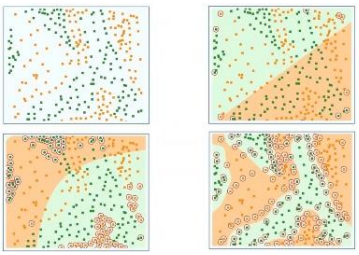
3.6.3.4 Support Vectors: Complementarity of Slackness

- **complementarity conditions (KKT theorem)**

$$\alpha_i[y_i(\mathbf{w}'\mathbf{x}_i - b) - 1 + d_i] = 0, \quad i \in \mathcal{M}$$
$$\mu_i(\alpha_i - \lambda) = 0, \quad i \in \mathcal{M}$$

- **the observations for which $0 < \alpha < \lambda$ are at distance $\frac{1}{\|w\|}$ from separating hyperplane and therefore lie on the canonical hyperplanes: support vectors**

If a constrain is exactly satisfied (constrain=0) his dual multiplier (which usually are equals to zero) can be greater than zero. So identifying the observation that has ALPHA greater than zero we identify support vector. As we said at the beginning support vector are the most important observation in SVM.



Linear, second degree polynomial, gaussian kernels example.

Having a large number of support vectors can be a symptom of overfitting.

3.6.3.5 Hyperparameters

Hyperparameters: LAMDA (generalizationVSmisclassification), Kernel function, Chosen Kernel hyperparameters (each have its own).

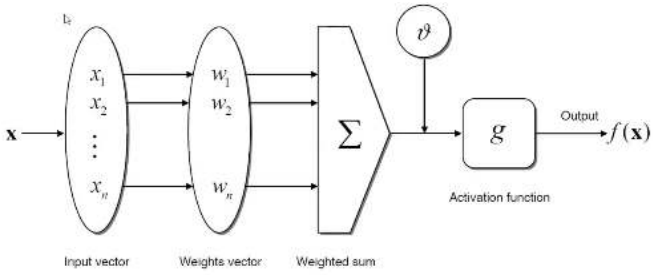
This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

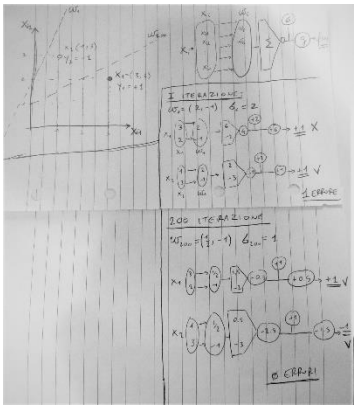
3.7 Artificial Neural Networks

They were invented in the 1950s, nevertheless neural networks were too slow to be used in real word situation. Thanks to algorithmic improvements and technological improvements they are now very popular. We will skip the mathematical detail of the partial derivatives methodology, since it is not necessary to understand the logic behind Neural Networks.

3.7.1 Perceptron

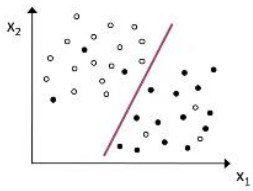


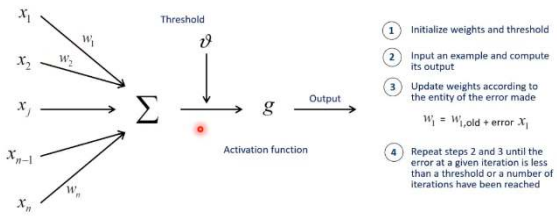
The input is represented by all the component of an X vector. The vector is multiplied component-wise by a vector of weights, then an intercept is subtracted (TETA) and then an activation function will map the linear combination into the set of the values assumed by the target variable (in case of binary classification it is usually the signum function).



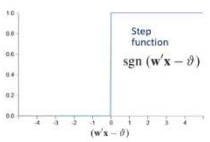
The separation is simply performed by a hyperplane and then the signum of the output is the prediction.

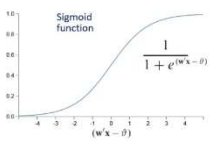
$$z = w_1x_1 + w_2x_2 + \cdots + w_nx_n - \vartheta = \mathbf{w}'\mathbf{x} - \vartheta$$



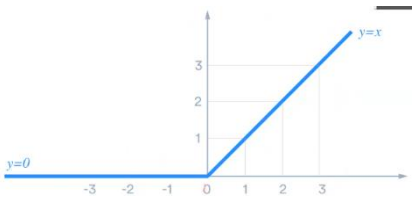


The algorithm to find the optimal separation hyperplane is gradient based. You consider one observation after the other: you start from random W and TETA values, you input the first observation and you update the weights based on the error you performed on the prediction. You repeat until a stopping rule is met.





The signum activation function just map whether a point lay on one side or on the other of the separating line. The problem is that is not differentiable in two points, that is why we can use the sigmoid instead.

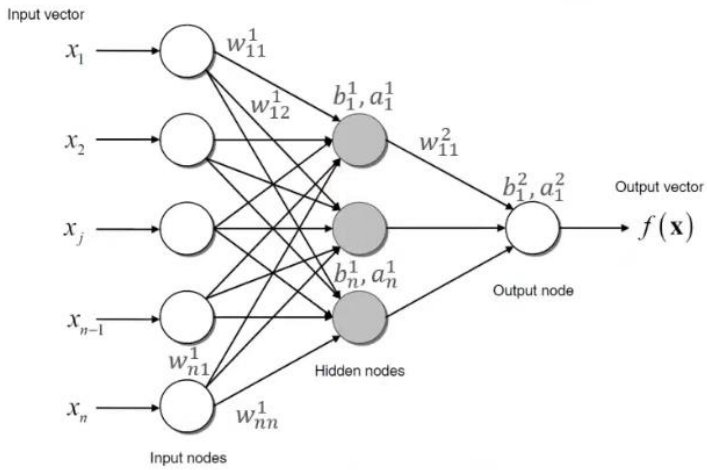


ReLu activation function. It is used mainly in internal nodes.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.7.2 Multi-Layer Neural Networks

Each hidden unit can be considered as a single perceptron.



1. Initialize weights and threshold.
2. Input an example, compute its output and the corresponding error.
3. Starting with the output layer propagate the error back to the previous layer and update the weights according to the "strength" of the connection.
4. Repeat steps 2 and 3 for all the training examples.

$$w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}$$

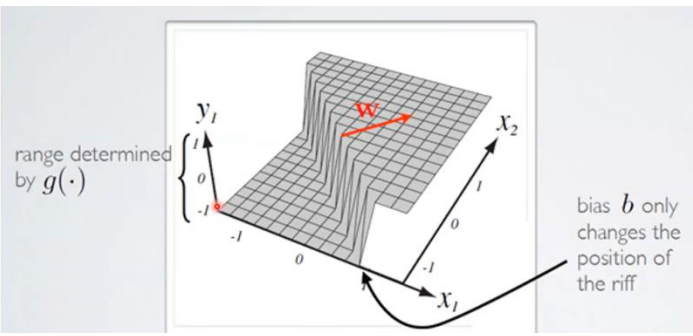
$a_i^l = \phi(\sum_n w_{ni}^l a_n^{l-1} + b_i^l)$, where ϕ is the element-wise activation function

Feed Forward: everything propagates in forward direction, from input to output.

The input nodes are always n (number of components of X). Then we have some hidden layers, in which the activation function is typically a ReLu. Every time we have a connection there is the relative weight.

After calculating the error, starting from the output node the algorithm propagates the error back to the other nodes based on the strength of the connection. From a technical point of view it uses chain rules of partial derivatives. The general formula for updating the weights: their previous value plus LAMBDA times error (actual value minus predicted value) times Xij.

3.7.3 Non-Linear Separation



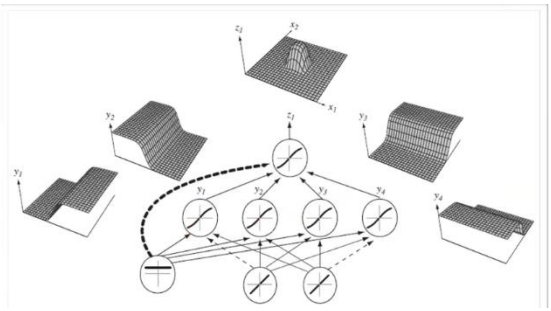
By varying the number of layers and the number of nodes in each layer we can perform non-linear separation.

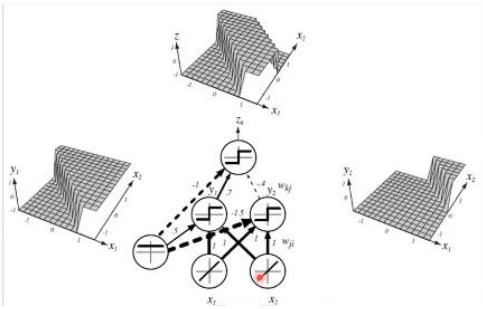
A single linear node performs a linear separation. The position depends on b and the slope depends on the W.

The combination of two or more linear separation becomes a non-linear separation.

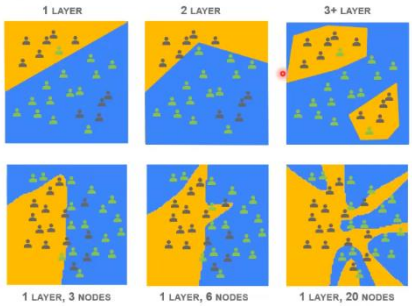
This document is available on studocu

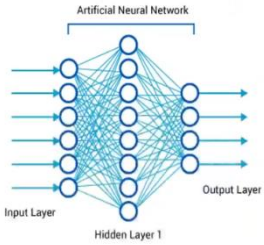
Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

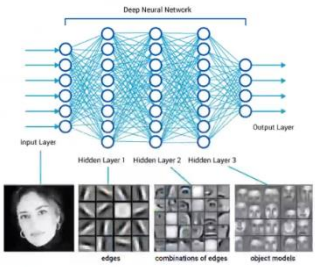




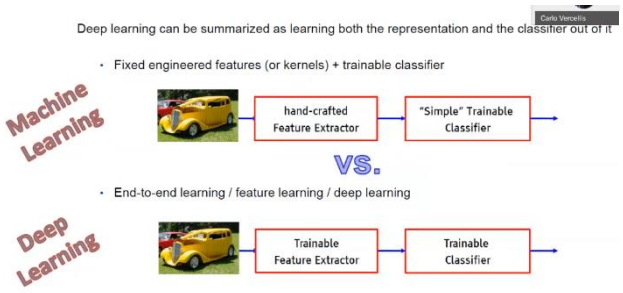
3.7.4 Deep Learning

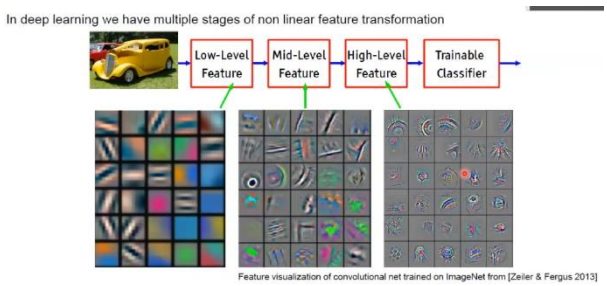






Deep learning refers to an artificial neural network with many (more than 5 up to 20 ore even 50) hidden layer. They work very good on unstructured data such as images and text, because each layer works as a feature extractor and then as a classifier (is saying each layer works on an increasing level of abstraction). In normal machine learning there were handcrafted features extractor, and the extracted features were fed into the classifier, in deep learning there are multiple stages on non-linear features extraction.





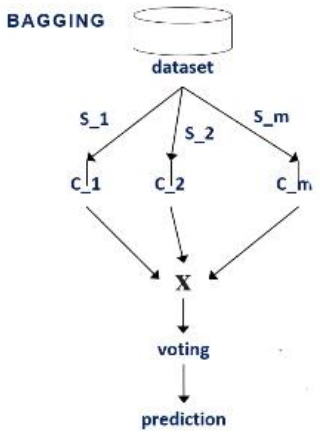
3.8 Ensemble Methods

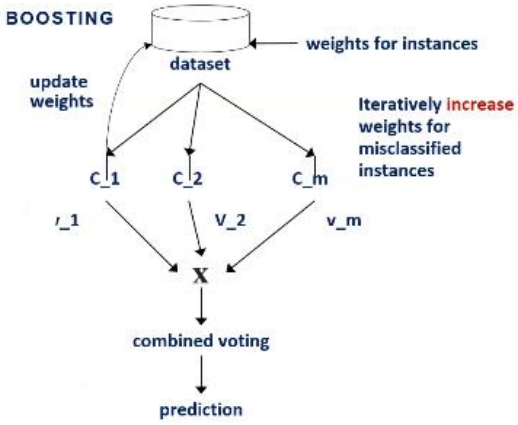
The idea is very simple, but they work very good in practice. The idea is that multiple classifiers can work better than a single one. In classification you do not average the results but apply majority voting principle. Usually you combine a large number of very simple classifiers.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

3.8.1 Bagging

better images on page 41





In bagging you divide the train data into random subsets, and you feed many times a classifier (could even be the same classifier with the same hyper parameters) but with different input data, so that you will obtain different results. Then for each new observation you will apply each classifier and through majority voting will derive final prediction. The goal is to have a more robust solution since it is based on many different independent models (and dataset).

3.8.2 Boosting

Boosting produces an ensemble model that is in general less biased than the weak learners that compose it. You assign a weight to each instance and to each weak classifier (starting from all weights=1), and update the instances weights by increasing the one of misclassified observations (more critical ones, will be handled by next weak classifier). One of the risks is to give more and more importance outlier! That is why at the end you apply combined voting, which is majority voting principle but allocating a higher weight to a better performing classifier.

3.8.3 Bagging: Random Forest

- Each classifier in the ensemble is a classification tree and is generated using a random selection of attributes at each node to determine the split.
- Forest-RI (random input selection): Randomly select, at each node, F attributes as candidates for the split at the node.
- Forest-RC (random linear combinations): Creates new attributes (or features) that are a linear combination of the existing attributes (reduces the correlation between individual classifiers).

RC actually generates oblique trees.

3.8.4 Boosting: AdaBoost

- Initially, all the weights of tuples are set the same (1/d)
- Generate k classifiers in k rounds. At round i,
  - Tuples from D are sampled (with replacement) to form a training set D_i of the same size
  - Each tuple's chance of being selected is based on its weight
  - A classification model M_i is derived from D_i and its error rate is calculated
  - If a tuple is misclassified, its weight is increased, otherwise it is decreased
- Error rate: $err(X_j)$ is the misclassification error of tuple X_j. Classifier M_i error rate is the sum of the weights of the misclassified tuples:
- The weight of classifier M_i's vote is $error(M_i) = \sum_{j}^{d} w_j \times err(\mathbf{X}_j)$

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

4. Regression

4.1 Structure of Regression Problems

Supervised learning applied to numerical target variables.

- dataset $\mathcal{D}$ contains m observations and $n+1$ attributes
- n independent attributes (explanatory, predictors) and 1 dependent attribute (target, response)
- observations $\mathbf{x}_i, i \in \mathcal{M}$ are points in a n dimensional space, the target attribute is denoted as y_i
- $\mathbf{X}$ is the $m \times n$ matrix of data, and $\mathbf{y}$ is the target vector
- Y, X_j are random variables, $f : \mathbb{R}^n \to \mathbb{R}$

$$Y = f(X_1, X_2, \ldots X_n).$$

As for classification, all categorical variables have been transformed into numerical ones via an orthonormal transformation (dummy variables) since all techniques (apart from regression trees) requires this transformation. Y and Xj are random variables. Regression is an approximation problem whereas classification is a separation problem.

4.1.1 Spurious Correlation

Mathematically the independent variable explains the Y but there is not any causal effect, you do not know why it works. You can use a model based on spurious correlation but with attention, since you do not know if the model will hold in the future.

4.1.2 Hypothesis Space

- linear

$$Y = w_1 X_1 + w_2 X_2 + \cdots + w_n X_n + b = \sum_{j=1}^{n} w_j X_j + b.$$

- quadratic $\quad Y = b + wX + dX^2 \qquad Z = X^2.$

$$Y = b + wX + dZ.$$

- exponential $\quad Y = e^{b+wX} \qquad Z = \log Y. \qquad Z = b + wX.$

It must be very simple in order to avoid overfitting (even more than in classification).

In most cases is possible to stay in the linear field thanks to a simple transformation (quadratic, exponential).

4.2 Least Square Regression

4.2.1 Simple Regression Problem

- deterministic model

$$Y = wX + b.$$

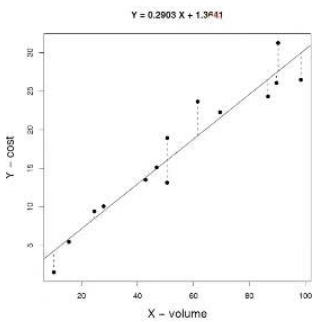
- probabilistic model

$$Y = wX + b + \varepsilon.$$

Volume X (ton)	Cost Y (€$\times 10^3$)	Volume X (ton)	Cost Y (€$\times 10^3$)
10.11	1.53	42.87	13.51
50.56	13.14	61.53	23.65
90.28	31.24	24.60	9.43
15.50	5.47	46.85	15.12
69.52	22.27	50.63	18.94

Single-variate, linear space of hypothesis.

From a deterministic model to a probabilistic one, with an EPSILON which is the random variation (noise, error, residual) from the deterministic model.



Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

- residuals

$$e_i = y_i - f(x_i) = y_i - wx_i - b, \quad i \in \mathcal{M}.$$

- least squares regression: minimize the sum of squared residuals

$$\text{SSE} = \sum_{i=1}^{m} e_i^2 = \sum_{i=1}^{m} [y_i - f(x_i)]^2 = \sum_{i=1}^{m} [y_i - wx_i - b]^2$$

The unknow quantities in the minimization problem are the W and b, not X and Y which are known. It is an unconstrained problem, quadratic (non-linear) in two variables. To solve it we take the partial derivatives and nullify them, there is one unique solution since it is a parabolic function.

$$\frac{\partial \text{SSE}}{\partial b} = -2 \sum_{i=1}^{m} [y_i - wx_i - b] = 0,$$

$$\frac{\partial \text{SSE}}{\partial w} = -2 \sum_{i=1}^{m} x_i [y_i - wx_i - b] = 0.$$

The system of two equation with two unknown quantities has a unique solution when the determinant is not zero.

The terms in the normal equation are one proportional to the variance of the independent variables (Xi^2) and two proportional to the average of the independent variables (Xi).

$$\begin{pmatrix} m & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} x_i^2 \end{pmatrix} \begin{pmatrix} b \\ w \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i y_i \end{pmatrix}$$

In the constant term the first is proportional to the average of the dependent variable (Yi) and the second to the covariance (XiYi).

$$\hat{w} = \frac{\sigma_{xy}}{\sigma_{xx}},$$
$$\hat{b} = \bar{\mu}_y - \hat{w}\bar{\mu}_x,$$

Variances and covariances are the key quantities.

The hat on W and b indicate these are the sample estimation of the real unknown quantities.

$$\bar{\mu}_x = \frac{\sum_{i=1}^{m} x_i}{m}, \qquad \bar{\mu}_y = \frac{\sum_{i=1}^{m} y_i}{m}.$$

- prediction

$$\hat{Y} = \hat{f}(X) = \hat{b} + \hat{w}X = \bar{\mu}_y + \frac{\sigma_{xy}}{\sigma_{xx}}(X - \bar{\mu}_x).$$

$$\sigma_{xy} = \sum_{i=1}^{m} (x_i - \bar{\mu}_x)(y_i - \bar{\mu}_y).$$
$$\sigma_{xx} = \sum_{i=1}^{m} (x_i - \bar{\mu}_x)^2,$$
$$\sigma_{yy} = \sum_{i=1}^{m} (y_i - \bar{\mu}_y)^2,$$

Sometimes is better to force the line to pass through the origin (it must be done only when from domain expertise you know it must pass from zero) imposing b=0.

$$\hat{w} = \frac{\sum_{i=1}^{m} x_i y_i}{\sum_{i=1}^{m} x_i^2},$$
$$\hat{b} = b = 0,$$

4.2.2 Multiple Linear regression

- probabilistic model

$$Y = w_1 X_1 + w_2 X_2 + \cdots + w_n X_n + b + \varepsilon.$$

Generalized to n-dimensional problem, using matrix notation. We include the intercept into the W vector.

$$\mathbf{e} = (e_1, e_2, \ldots, e_m)$$
$$\mathbf{w} = (b, \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$$

We extend the matrix X by a vector with all components=1 (one 1 to each observation) to re-write the formula this way.

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \mathbf{e}.$$

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

The transpose of a product is the product of the transpose with reverse order: (XW)' = W'X'

- sum of squared residuals $$\mathrm{SSE} = \sum_{i=1}^{m} e_i^2 = \|\mathbf{e}\|^2 = \sum_{i=1}^{m} (y_i - \mathbf{w}'\mathbf{x}_i)^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})'(\mathbf{y} - \mathbf{X}\mathbf{w}).$$
- normal equation $$\mathbf{X}'\mathbf{X}\mathbf{w} = \mathbf{X}'\mathbf{y}.$$
- null partial derivatives $$\frac{\partial\, \mathrm{SSE}}{\partial \mathbf{w}} = -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\mathbf{w} = \mathbf{0}.$$
- minimum point $$\mathbf{w} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}.$$

If X'X has determinant different than zero it has an inverse, and we can calculate the minimum point.

There are some problems in this technique: it is an ill-posed problem. This is because you are minimizing the error on the past data, completely disregarding generalization, with a high risk of overfitting. Ridge and Lasso regression try to solve this problem.

From a mathematical point of view, the calculation of the inverse can be critical, when in the covariance matrix (X'X) the eigen values are very disperse (the highest many times greater than the smallest), because in that case the matrix times its inverse is not equal to the identical matrix.

- values predicted by model $$\boxed{\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{w}} = (\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{y} = \mathbf{H}\mathbf{y}.}$$

You put the Y out of the parentheses in order to highlight the hat matrix. $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$

We can see that the predictions are a matrix (H) times the past data on which we trained.

The residuals: $$\mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}.$$

It is nice to note that the algorithms that performs least square regression mainly do linear and matrix algebra.

4.3 Assumptions on the Residuals

4.3.1 Normal, Zero Mean, Constant Variance

- random variable ε should follow a normal distribution of mean 0 and constant variance

$$\mathrm{E}(\varepsilon_i|\mathbf{x}_i) = 0,$$
$$\mathrm{Var}(\varepsilon_i|\mathbf{x}_i) = \sigma^2$$

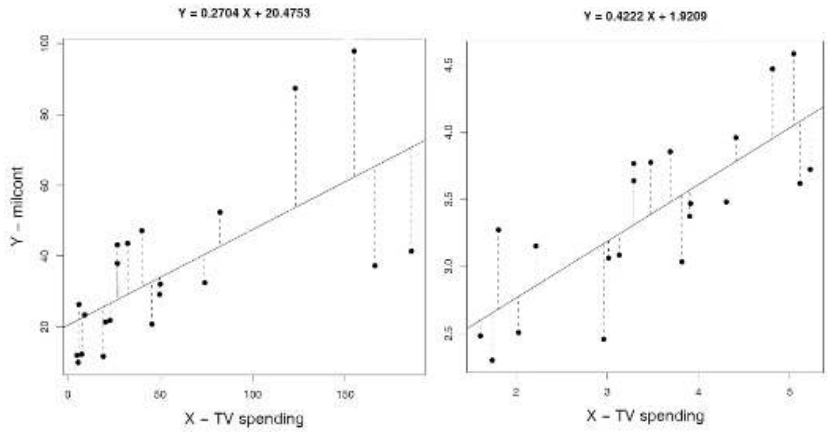
4.3.2 Independence of the Residuals

The first assumption states that residuals are pure noise. If not, some more variance could be captured by the model.

- residuals ε_i e ε_k should be independent
- estimate of σ $$\bar{\sigma}^2 = \frac{\mathrm{SSE}}{m-n-1} = \frac{\sum_{i=1}^{m}(y_i - \mathbf{w}'\mathbf{x}_i)^2}{m-n-1} = \frac{\mathbf{y}'(\mathbf{I}-\mathbf{H})\mathbf{y}}{m-n-1},$$
- if standard deviation σ is constant we have homoscedasticity, otherwise heteroscedasticity

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

4.3.3 Heteroscedasticity



Increasing variance (very common, larger error on larger values). It is not acceptable, but you can simply adopt a logarithmic scale on both variables to solve the problem.

4.4 Ridge Regression

- the estimation of matrix $(\mathbf{X}'\mathbf{X})^{-1}$ can be critical (insufficient number of observations, multi-collinearity): ill-posed problem
- limit the width of the hypothesis space F (regularization theory)

$$\min_{\mathbf{w}} RR(\mathbf{w}, \mathcal{D}) = \min_{\mathbf{w}} \lambda \|\mathbf{w}\|^2 + \sum_{i=1}^{m} (y_i - \mathbf{w}'\mathbf{x}_i)^2$$
$$= \min_{\mathbf{w}} \lambda \|\mathbf{w}\|^2 + (\mathbf{y} - \mathbf{X}\mathbf{w})'(\mathbf{y} - \mathbf{X}\mathbf{w})$$

Useful to avoid overfitting.

Very similar to support vector machines as a concept. You consider into the objective function the error on the past data and the generalization capability in the form of the norm of W. LAMBDA is the hyper parameter balancing the two.

It works, it is demonstrated that the inverse of this new problem is always well defined, so it is not ill-posed.

4.5 Lasso regression

- instead of L_2 norm, use L_1 norm

$$\min_{\mathbf{w}} LR(\mathbf{w}, \mathcal{D}) = \min_{\mathbf{w}} \lambda |\mathbf{w}| + \sum_{i=1}^{m} (y_i - \mathbf{w}'\mathbf{x}_i)^2$$
$$= \min_{\mathbf{w}} \lambda |\mathbf{w}| + (\mathbf{y} - \mathbf{X}\mathbf{w})'(\mathbf{y} - \mathbf{X}\mathbf{w})$$

It introduces non differentiability, but sometimes work better.

4.6 Generalized Linear Models

- Functions g_h represent any set of bases, such as polynomials, kernels and other groups of nonlinear functions

$$Y = \sum_h w_h g_h(X_1, X_2, \ldots, X_n) + b + \varepsilon$$

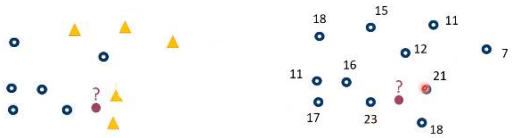
- Coefficients w_h and b can be determined through the minimization of the sum of squared errors. Function SSE in this formulation is more complex than for linear regression, solution of the minimization problem more difficult

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

4.7 K-Nearest Neighbors

Classification: majority voting among K nearest **Regression:** average of K nearest



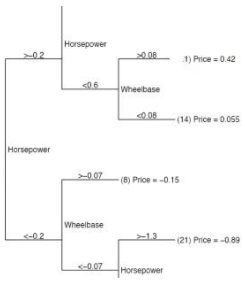
Average is the most popular, but any measure of central tendency is ok.

4.8 Regression Trees

- replace Gini, entropy or misclassification with

$$S = \sum_{c \in \text{leaves}(T)} \sum_{i \in C} (y_i - m_c)^2$$

- where the prediction is $m_c = \frac{1}{n_c} \sum_{i \in C} y_i$



The prediction will be the average of the target values of the observations in one leaf.

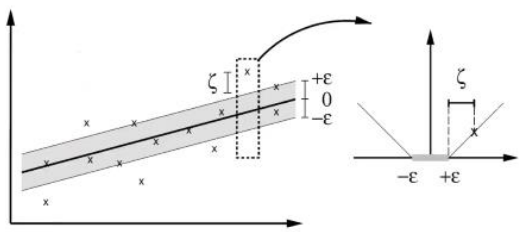
The confusion index will be replaced by the sample variance (choose the splitting rules which produces the nodes with minimum variance, since it means observation are closer together in the target value).

Random forests usually work really well.

4.9 Support Vector Regression

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) \\ \text{subject to} & \left\{ \begin{array}{lcl} y_i - \langle w, x_i \rangle - b & \leq & \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i & \leq & \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq & 0 \end{array} \right. \end{array}$$

$$|\xi|_\varepsilon := \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases}$$



C is the new LAMBDA.

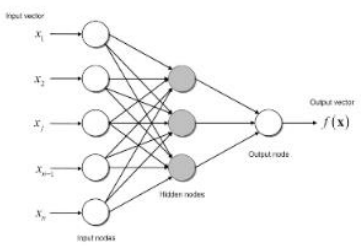
Instead of the separation hyperplane we are looking for the fitting hyperplane. As opposed as the separation margin, here there are two hyperplanes parallel to the fitting one, but the points falling inside that grey region are not recognized as errors. EPSILON is an hyperparameter.

Then you re-formulate the problem and then solve it as we did for support vector.

4.10 Neural Network

- linear activation function for the output node

$$g(\mathbf{w}'\mathbf{x} - \vartheta) = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n - \vartheta = \mathbf{w}'\mathbf{x} - \vartheta$$



NN are completely the same for classification and for regression.

The only difference is that the activation function for the output nodes (signum or sigmoid for classification), is just the linear combination in the last node.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

4.11 Assessing Performance of the Model

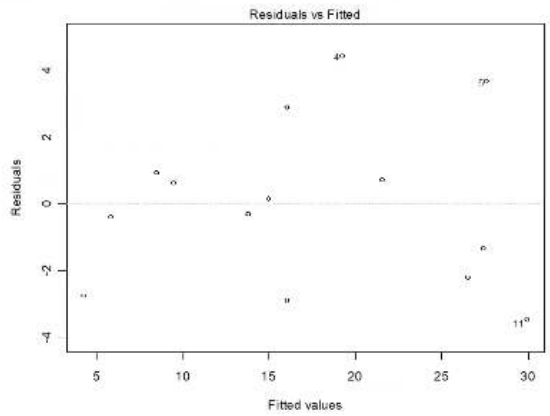
4.11.1 Normality and Independence of the Residuals

It applies to all model: least squares, KNN, NN ecc…

- goodness-of-fit test (chi-squared, Kolmogorov-Smirnov)
- graphical analysis
  - scatterplot residuals vs. fitted
  - scatterplot standardized residuals vs. fitted
  - qq-plot of the residuals

It is not a measure of the performance, but it is a condition that has to be met to consider the model valid and feasible. To check normality there are many test (chi-squared is valid for any distribution, K-S is only to check normality). If the test rejects the normality hypothesis, you have to change model, changing the space of hypothesis, or because you do not have enough knowledge of the process (need more explanatory variables).

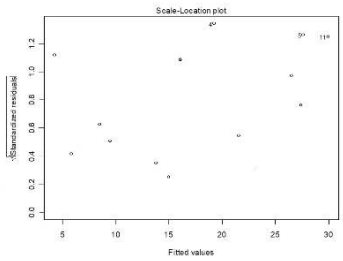
4.11.1.1 Residuals VS Fitted



You should not see any pattern

Most usual situation is a conic shape, increasing residuals when increasing the fitted value.

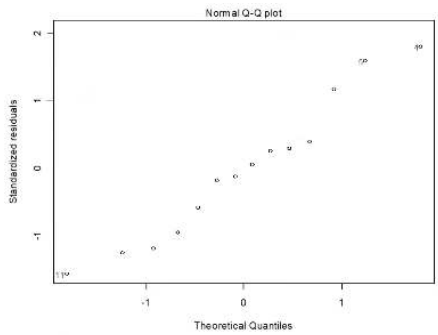
4.11.1.2 Standardized Residuals VS Fitted



Res-mean/std.

Square root of standardized residuals.

4.11.1.3 QQ-Plot



Observations should form a straight line.

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

4.11.2 Significance of the Coefficients

Only for least square regression.

- regression coefficients $\hat{\mathbf{w}}$ represent an estimate of w
- covariance matrix of the estimator $\mathbf{V} = \mathrm{cov}(\hat{\mathbf{w}}) = \tilde{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$
- confidence intervals $w_j \pm t_{\alpha/2}\sqrt{v_{jj}}$,
- for simple regression

$$w \pm t_{\alpha/2}\frac{\tilde{\sigma}}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)^2}},$$

$$b \pm t_{\alpha/2}\frac{\tilde{\sigma}}{\sqrt{n}}\sqrt{1 + \frac{n\bar{\mu}_x}{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)^2}}.$$

(null hypothesis) $H_0 : w = 0$, (alternative hypothesis) $H_a : w \neq 0$

predictor	value	standard error	t-value	Pr > |t|
(intercept)	1.36411	1.48944	0.916	0.3780
volume	0.29033	0.02423	11.980	< 0.0001

The obtained coefficients are just an estimate of the true parameters, based on the sample, are they good?

The covariance of the estimator W is proportional to the variance of the residuals sigma square hat.

We can so build a confidence interval (with T-Student because it could be a small sample).

We can also use a hypothesis test to check whether the confidence interval includes the zero or not. If it does the model is not trustworthy since we do not even know if one variable has a positive or a negative effect on target.

4.11.2.1 T-Student Test

We check weather one component of W is equal to 0.

In the example the coefficient for the volume is very close to zero, so it may seem it is not significant. But the width of the confidence interval heavily depends on the standard deviation of the errors (since T-Student for ALPHA 0.05 is around 2). The standard error of the W is very small too. The t value in the table states how many standard errors the W value is far from zero. The p-value tells us the probability that the real W is 0.

4.11.2.2 ANOVA Test

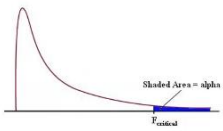
- df: *n* degrees of freedom
- sum of sq: $\mathrm{RSS}_{reg} = \sum_{i=1}^{m}(\hat{y}_i - \bar{\mu}_y)^2 = 933.18.$
- df: *m-n-1* degrees of freedom
- df: *m-1* degrees of freedom
- sum of sq:

$$\mathrm{RSS}_{tot} = \sum_{i=1}^{m}(y_i - \bar{\mu}_y)^2 = 1011.20.$$

predictor	df	sum of sq.	mean sq.	F-value	Pr > F
volume	1	933.18	933.18	143.53	< 0.0001
residuals	12	78.02	6.50		
total	13	1011.20	77.79		

- the aim of regression models is to explain through predictive variables most part of variance inherent in dependent variable, leaving aside pure random fluctuation – residuals
- If this goal is achieved, one expects sample variance of residuals significantly smaller than sample variance of response variable
- if the residuals have normal distribution, the following ratio follows an *F* distribution with *n* e *m-n-1* degrees of freedom

$$F = \frac{\mathrm{RSS}_{\mathrm{reg}}/n}{\mathrm{SSE}/(m-n-1)}$$



RSSreg is calculated as the variance but instead of using the actual value of Y we use the predicted value. In this way we calculate the variance which is actually explained by the model.

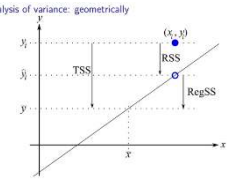
RSStot is the actual variance of the dependent variable of the data.

These two quantities should be as close as possible, leaving the SSE as small as possible.

Mean square is the SSE divided by the degrees of freedom.

The F value must be big enough, leaving a small area on the right (small p/value).

The analysis of variance: geometrically



Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

4.11.3 Determination Coefficient

- determination coefficient $R^2 = \frac{\text{RSS}_{reg}}{\text{RSS}_{tot}} = \frac{\sum_{i=1}^{m}(\hat{y}_i - \bar{\mu}_y)^2}{\sum_{i=1}^{m}(y_i - \bar{\mu}_y)^2}$.
- in the example $R^2 = 0.9228$.
- adjusted coefficient $R^2_{adj} = 1 - (1 - R^2)\frac{m-1}{m-n-1}$.
- in the example $R^2_{adj} = 0.9164$.

Ok for any type of technique (because any technique produces a prediction and you only need predictions in the formula).

The closest to 1 the better, since it is the percentage of variation in the data explained by the model.

In Radj we have to take in account the ratio between the number of explanatory variables N and the number of observation M (preventing overfitting).

4.11.4 Regression Coefficient

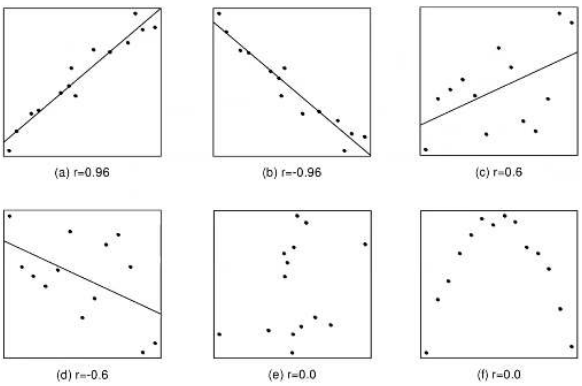
statistics	value
residual standard error	2.5500
multiple R-squared	0.9228
adjusted R-squared	0.9164
regression coeff.	0.9606

When there is more than one independent variable (multi-variate model), the regression coefficient is defined as the square root of R.

On simple models we can use the classic definition, and the two definitions are consistent.

4.11.5 Linear Correlation Coefficient

$$r = \text{corr}(\mathbf{y}, \mathbf{x}_i) = \frac{\sigma_{xy}}{\sqrt{\sigma_{xx}\sigma_{yy}}} = \frac{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)(y_i - \bar{\mu}_y)}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)^2 \sum_{i=1}^{m}(y_i - \bar{\mu}_y)^2}}$$



- if r > 0, then X and Y are concordant.
- if r < 0, then X and Y are discordant.
- finally if r is close to 0, there is non linear relationship between X e Y.

4.12 Multi-Collinearity of the Independent Variables

It is a problem for least square regression. It is one of the main reasons why the matrix that has to be inverted (X'X) could be hard to invert. It means two independent variables are correlated. You can check calculating the pairwise correlation coefficient and if it is lower than a threshold, they are ok.

- variance inflation factor: values greater than 5 point to the existence of multi-collinearity

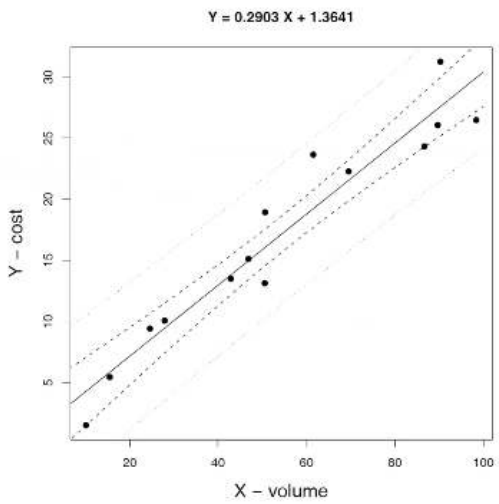
$$\text{VIF}_j = \frac{1}{1 - R_j^2}$$

But you cannot check whether a single variable is correlated to a group of other variables, so you can check the VIF index. You build another least regression model taking as a target one independent variable at the time and as explanatory variables the remaining N-1. If the VIF is greater than 5, it means that R^2 for that model is greater than 0.8. It means that this variable is explained enough by the remaining ones (at least the 80% of its variance).

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

4.13 Confidence and Prediction Limits



- prediction associated to a new observation $\hat{y} = \hat{\mathbf{w}}\mathbf{x}$
- its variance is $\mathrm{Var}(\hat{y}) = \mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}\bar{\sigma}^2,$
- for simple regression the confidence limit for E[*Y*] is

$$\hat{y} \pm t_{\alpha/2}\bar{\sigma}\sqrt{\frac{1}{m} + \frac{x - \bar{\mu}_x}{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)^2}},$$

- whereas the prediction interval for *Y* is

$$\hat{y} \pm t_{\alpha/2}\bar{\sigma}\sqrt{1 + \frac{1}{m} + \frac{x - \bar{\mu}_x}{\sum_{i=1}^{m}(x_i - \bar{\mu}_x)^2}}.$$

Only least square regression.

Sigma bar is the standard deviation of the residuals.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

5. Clustering

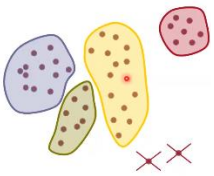
5.1 General Concepts

Objective ⇒ homogeneous groups of points (CLUSTERS):

- points within a cluster should be similar
- points from different clusters should be dissimilar

Can be used for:

- providing meaningful interpretations
- preliminary grouping of points
- detecting outliers
- selecting landmarks



Several application domains...

- **Marketing**
  partition consumers into market segments (better understand the relationships between different groups of consumers/potential customers)
- **Social network analysis**
  recognize communities (hubs) within large groups of people
- **Genomics**
  build groups of genes with related expression patterns (often such groups contain functionally related proteins)
- **Social science**
  identify areas (hot spots) where there are greater incidences of similar types of crime to manage law enforcement resources

Unsupervised learning.

Preliminary grouping of point could be done before a classification/regression problem to apply different classifier to each cluster.

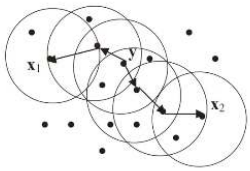
5.1.1 Taxonomy of Clustering Methods

Based on the logic used for building clusters we can have:

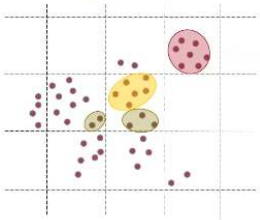
- PARTITION METHODS
  the data set is divided into a pre-fixed number of clusters
- HIERARCHICAL METHODS
  perform several partitions based on a tree structure
- DENSITY-BASED METHODS
  look at the number of points lying within the neighborhood of each point
- GRID METHODS
  perform a preliminary partition based on a grid structure

We will only study the first two methods. The fourth is a preliminary method to simplify the problem before applying one of the other three methods.

density-based methods



grid methods

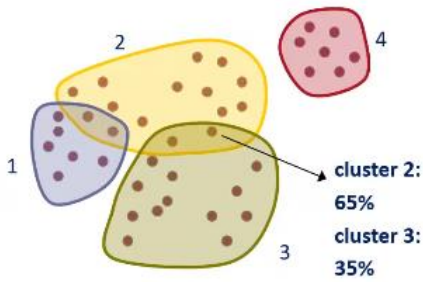


Grid methods are useful when there is very big dataset (more than a billion, for example SM analysis). The grid divides the dataset into subgroup and then build clusters into the cells of the grid. As a final step, we can aggregate by merging two cluster which appear close in two adjacent grid cells.

Density based methods: You start from a point and you build a neighborhood of the point and include all the observation that fall inside.

Based on the logic used for assigning the points we can have:

- EXCLUSIVE METHODS
- FUZZY METHODS
- COMPLETE METHODS
- PARTIAL METHODS



Exclusive Methods: Each point belongs at one cluster or at the other.

Fuzzy Methods: Not clear cut, some points appertain to two cluster with given probabilities.

Complete Methods: Each observation belongs to a cluster.

Partial Methods: Accepts that some observation don't belong to any cluster.

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

5.1.2 Assessing Performance

General requirements:

- **FLEXIBILITY** ⇒ **numeric and categorical attributes**
- **ROBUSTNESS** ⇒ **stability of the clusters (noise)**
- **EFFICIENCY** ⇒ **small computing time**

A huge number of possible partitions...

... clustering is *NP*- hard for $K \geq 3$

⇓

Most methods are heuristic in nature!

$$\frac{1}{K!}\sum_{h=1}^{K}\binom{K}{h}h^m \quad \text{possible combinations}$$

For numerical variables, the notion of distance is clear. Categorical are either equals or different.

We want the growth of computing time to grow linearly and not exponentially with the growth of the dataset (efficiency).

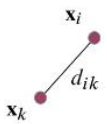
Unfortunately, there is a huge number of possible clusters, as they grows exponentially with the number of observations (h^m). For k>2 (k=number of clusters) it is an NP-hard problem and is always exponential, so it must be solved heuristically.

5.1.3 Affinity Measures

DISTANCE MATRIX

$$D = [d_{ik}] = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1,m-1} & d_{1m} \\ & 0 & \cdots & d_{2,m-1} & d_{2m} \\ & & \cdots & \vdots & \vdots \\ & & & 0 & d_{m-1,m} \\ & & & & 0 \end{bmatrix}$$

$$d_{ik} = \text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \text{dist}(\mathbf{x}_k, \mathbf{x}_i), \quad i, k \in \mathcal{M}$$



SIMILARITY MEASURE

$$s_{ik} = \frac{1}{1 + d_{ik}} \qquad s_{ik} = \frac{d_{max} - d_{ik}}{d_{max}}$$

There are many different definitions of similarity. Max similarity is one, minimum is zero.

5.1.4 Distance Measures

In Minkowsky distance q is almost never greater than 8 (1 is Manhattan, 2 is Euclidian).

EUCLIDEAN DISTANCE:

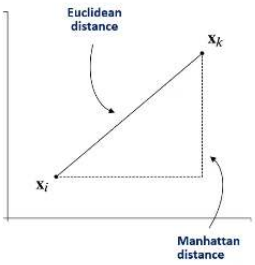
$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sqrt{\sum_{j=1}^{n}(x_{ij} - x_{kj})^2}$$

MANHATTAN DISTANCE:

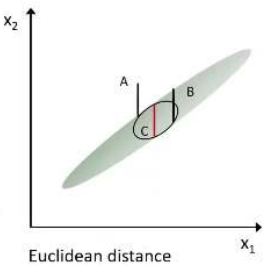
$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sum_{j=1}^{n}|x_{ij} - x_{kj}|$$

MINKOWSKI DISTANCE:

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sqrt[q]{\sum_{j=1}^{n}|x_{ij} - x_{kj}|^q}$$

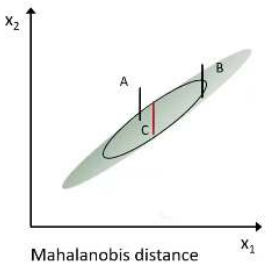


MAHALANOBIS DISTANCE: $\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sqrt{(\mathbf{x}_i - \mathbf{x}_k)\mathbf{V}^{-1}(\mathbf{x}_i - \mathbf{x}_k)'}$



Euclidean distance

› the set of points equidistant from a given location is a sphere



Mahalanobis distance

› stretches the sphere to account for correlation among variables

Mahalanobis distance is very similar to PCA.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

5.1.5 Affinity Measures for Binary Attributes

Contingency table

		point $\mathbf{x}_k$		
		0	1	totale
point $\mathbf{x}_i$	0	p	q	$p+q$
	1	u	v	$u+v$
	totale	$p+u$	$q+v$	n

Every categorical variable can be transformed to binary variables.

Let's assume all the n attributes are binary and build a contingency table.

COEFFICIENT OF SIMILARITY
(symmetric attributes)

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{q+u}{p+q+u+v}$$

The coefficient of similarity make sense when the attributes are symmetric (50% are 0 and 50% are 1).

JACCARD DISTANCE
(asymmetric attributes)

When is not balanced, here is the Jaccard distance.

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{q+u}{q+u+v}$$

There are measures also for nominal and ordinal attributes but is preferrable to transform them to dummies and use the binary notion of distance.

Nominal attributes

- **Jaccard coefficient**
- **extension of the coeff. of similarity** $\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{n-f}{n}$

Ordinal attributes

1) **standardization**
2) **distance measures for numeric attributes** $x'_{ij} = \frac{x_{ij}-1}{H_j-1}$

5.1.6 Affinity Measures: General Case

We are actually using the Manhattan distance and the similarity coefficient.

In case of both numeric and categorical variables:

- **define a binary index δ_{ikj} taking the value 0 when**
  - **at least x_{ij} or x_{kj} is missing**
  - **the attribute is binary and asymmetric $x_{ij} = x_{kj} = 0$**
- **define the coefficient Δ_{ikj} which indicates how much the attribute contributes to the similarity of points $\mathbf{x}_i$ and $\mathbf{x}_k$**

  - if the attribute is binary or nominal, we set $\Delta_{ikj} = 0$ if $x_{ij} = x_{kj}$, otherwise $\Delta_{ikj} = 1$

  - if the attribute is numeric, we set $\Delta_{ikj} = \frac{|x_{ij} - x_{kj}|}{\max_l x_{lj}}$

- **compute the similarity coefficients between $\mathbf{x}_i$ and $\mathbf{x}_k$**

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{\sum_{j=1}^{n} \delta_{ikj} \Delta_{ikj}}{\sum_{j=1}^{n} \delta_{ikj}}$$

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

5.2 Partition Methods

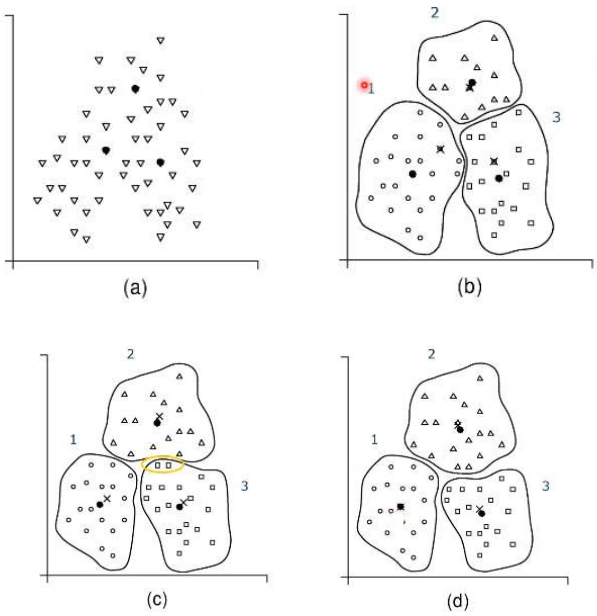
PARTITION METHODS: general framework

- initialization ⇒ points are divided into K non-empty groups (usually exhaustive and mutually exclusive)
- iteration ⇒ points are reassigned with the aim of improving the quality of the partition
- stop ⇒ no points are further reassigned (other stopping criteria)

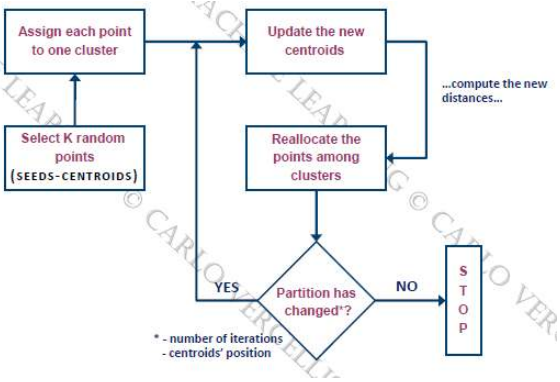
K is an hyperparameter. You try different values of K and check the results.

Partition methods are greedy!

5.2.1 K-Means Algorithm



The first three centroids are k random observation. Then they are just the centers of gravity in the space, not necessarily an observation.



5.2.1.1. Strengths and Weaknesses

STRENGTH AND WEAKNESS

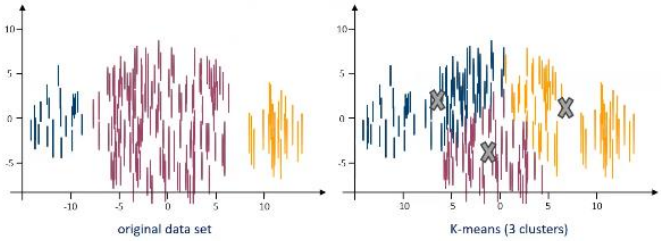
- relatively efficient: O(m*K*t*d) [m points, K clusters, t iterations, d variables]
- it converges for common similarity measures mentioned above (most of the convergence happens in the first few iterations)
- need to specify the number of clusters in advance
- often terminates at a local optimum (greedy algorithm)
- results may vary based on random seed selection, i.e. clusters may be different from one run to another (try out multiple starting points, FFT algorithm)
- applicable only when mean is defined...what about categorical data?
- unable to handle noisy data and outliers

Try multiple starting points since results may vary. Not good in handling categorical and binary data.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

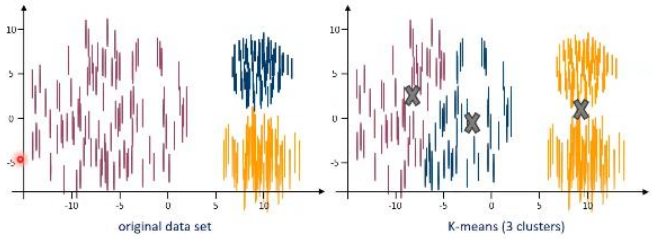
STRENGTH AND WEAKNESS

- it may have problems when (natural) clusters:
  - are of different sizes



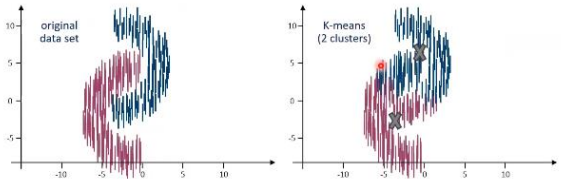
STRENGTH AND WEAKNESS

- it may have problems when (natural) clusters:
  - are of different sizes
  - have different densities



STRENGTH AND WEAKNESS

- it may have problems when (natural) clusters:
  - are of different sizes
  - have different densities
  - have non-convex shapes

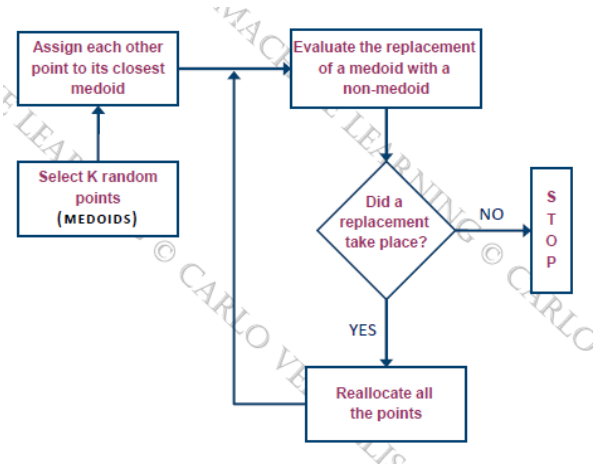


Pre and post-processing practical suggestions

PRE-PROCESSING
- **normalize (or standardize) the data**
- **eliminate outliers**

POST-PROCESSING
eliminate small clusters that may represent outliers
- **split "loose" clusters, i.e. clusters with relatively high SSE**
- **merge clusters that are "close" and have relatively low SSE**

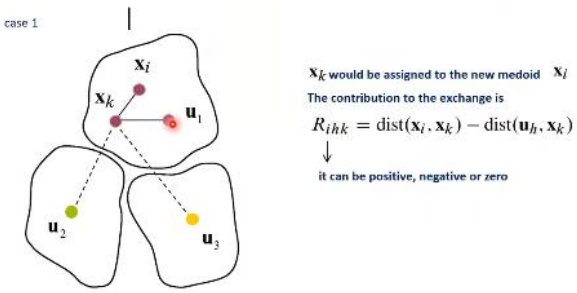
5.2.2 K-Medoids Algorithm



Medodids are always points belonging to the original dataset.

Each medoid's replacement is evaluated as follows:

- consider all the pair of points $(\mathbf{x}_i, \mathbf{u}_h)$
- evaluate how a third point $\mathbf{x}_k$ contribute to the exchange between $\mathbf{x}_i$ and $\mathbf{u}_h$

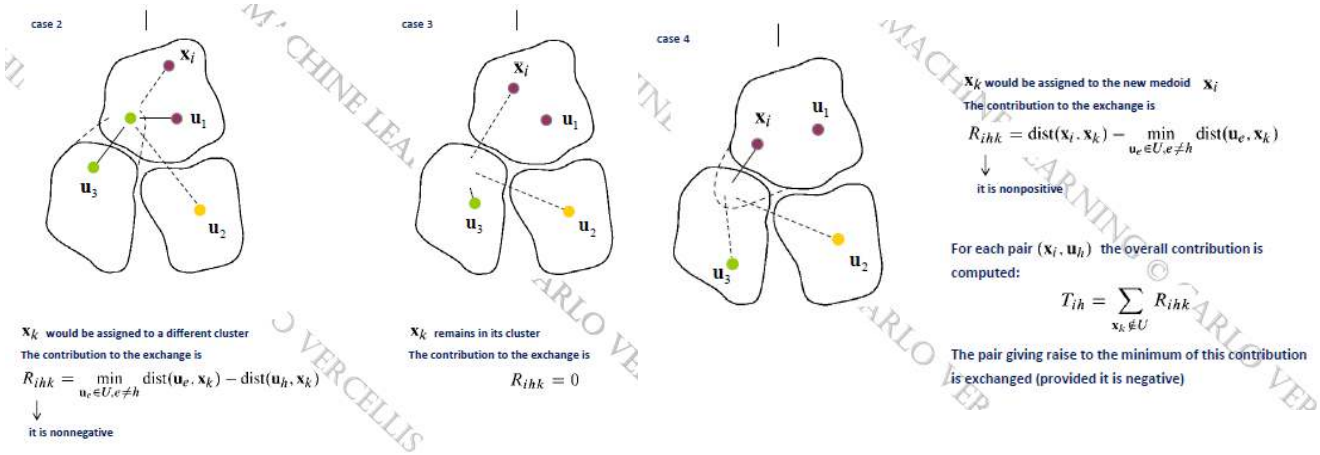


X are non-medoids, U are medoids.

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

It is an advantage when contribution is negative.



5.3 Hyerarchical Methods

HIERARCHICAL METHODS

- are based on a tree structure (dendogram)
- use the distances among points to derive clusters merging or splitting
- do not require the number K of clusters as an input

AGGLOMERATIVE ALGORITHMS (bottom-up techniques)

- initially each point represents a single cluster
- iteratively the two clusters with the minimum distance are merged together and the proximity matrix (distances among all clusters) is updated
- all points are comprised in a single cluster ⇒ stop

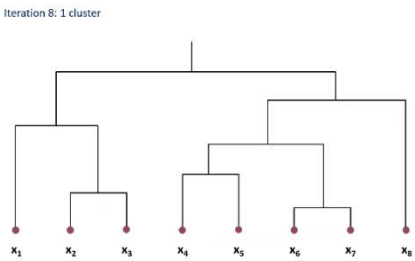
DIVISIVE ALGORITHMS (top-down techniques)

- initially all points are comprised in a single cluster
- iteratively choose a cluster and split it so to obtain two clusters with the maximum distance
- each point represents a single cluster ⇒ stop

Now we also need a measure of distance between cluster.

The divisive algorithms are too slow, so are not used in real life.

5.3.1 Agglomerative Methods



5.3.1.1 Measures of Distance Between Clusters

At each iteration the two clusters with the minimum distance are merged

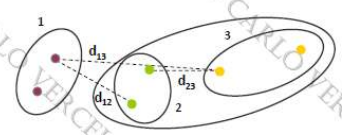
Several metrics can be used to evaluate the proximity (distance) between a pair of clusters

MINIMUM DISTANCE (SINGLE LINKAGE)

The distance of two clusters is based on the two most similar (closest) points in the different clusters

$$\mathrm{dist}(C_h, C_f) = \min_{\substack{\mathbf{x}_i \in C_h \\ \mathbf{x}_k \in C_f}} \mathrm{dist}(\mathbf{x}_i, \mathbf{x}_k)$$

- sensitive to noise and outliers
- biased towards elliptical clusters



Very simple and intuitive, but very sensible to outliers.

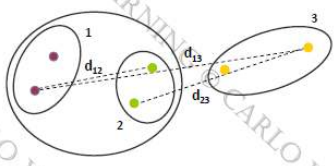
Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

MAXIMUM DISTANCE (COMPLETE LINKAGE)

The distance of two clusters is based on the two least similar (most distant) points in the different clusters

$$\mathrm{dist}(C_h, C_f) = \max_{\substack{x_i \in C_h \\ x_k \in C_f}} \mathrm{dist}(\mathbf{x}_i, \mathbf{x}_k)$$

- less sensitive to noise and outliers
- tends to break large clusters
- biased towards globular clusters

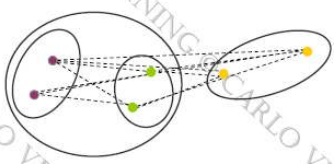


AVERAGE DISTANCE

The distance of two clusters is the average of the sum of the pairwise distances of the points in the two clusters

$$\mathrm{dist}(C_h, C_f) = \frac{\sum_{\mathbf{x}_i \in C_h} \sum_{\mathbf{x}_k \in C_f} \mathrm{dist}(\mathbf{x}_i, \mathbf{x}_k)}{\mathrm{card}\{C_h\}\,\mathrm{card}\{C_f\}}$$

- less sensitive to noise and outliers
- biased towards globular clusters



WARD'S MINIMUM VARIANCE METHOD

It minimizes the total within-cluster variance

- at the initial step all clusters are singletons
- compute the squared Euclidean distances among all the points
- compute the increase in total within-cluster variance for all possible merging

  Let d_{ik} d_{ij} and d_{jk} be the pairwise distances between the clusters C_i, C_j and C_k

  Let $d_{(ij)k}$ the distance (within-cluster variance) between the new cluster $C_i \cup C_j$ and C_k, where $d_{(ij)k} = \alpha_i \cdot d_{ik} + \alpha_j \cdot d_{jk} + \beta \cdot d_{ij} + \gamma\,|d_{ik} - d_{jk}|$

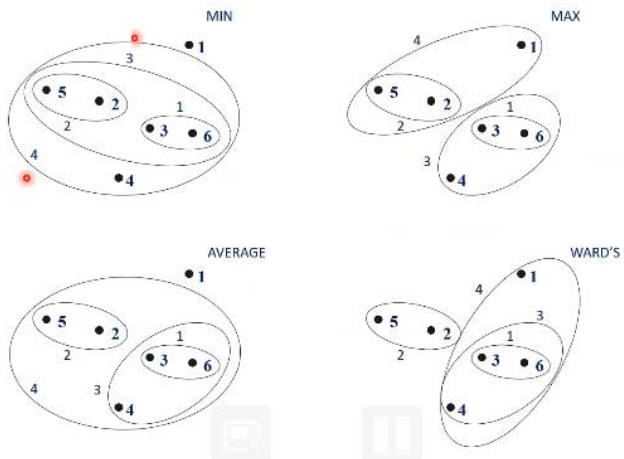
  Parameters depend on clusters size.

- merge the two clusters leading to the minimum increase of within-cluster variance

- less sensitive to noise and outliers
- biased towards globular clusters

Similar to regression trees, instead of splitting merges cluster in a way that variance is minimum.

When you perform a merging, the within cluster variance increase. You choose the one that increases the less.

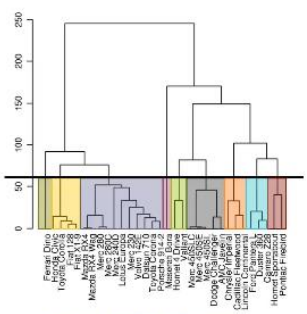


The choice of the notion of distance within clusters is an hyperparameter.

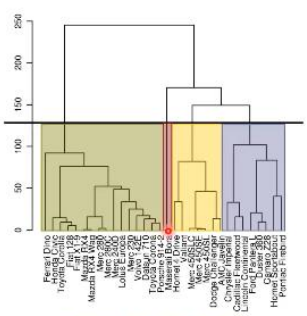
Furthermore, you can also choose different measures of distance within points (Euclidian, Manhattan) which is a second hyperparameter.

Depending on where you perform the cut you obtain a different number of clusters.

Any desired number of clusters can be obtained by "cutting" the dendogram at the proper level



average Euclidean distance
9 clusters cut



average Euclidean distance
4 clusters cut

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

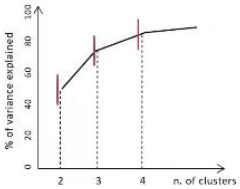
5.4 Clustering Validity

5.4.1 Number of Clusters

In aggregation methods you have to decide the number of clusters to obtain.

Alternative methods are available...

- **RULE OF THUMB** $K \approx \left(\frac{m}{2}\right)^{1/2}$
- **ELBOW METHOD**
  The percentage of variance explained* is set as a function of the number of clusters
  *Is the ratio of the within-cluster variance to the total variance



- **SILHOUETTE-BASED METHOD**
  Choose the number of clusters giving rise to the largest average silhouette

5.4.2 Internal Measures

Internal measures: used to measure the goodness of a clustering structure independently of external information

- **COHESION** of each cluster
  measures how closely related are objects in a cluster

$$\text{coes}(C_h) = \sum_{\substack{x_i \in C_h \\ x_k \in C_h}} \text{dist}(\mathbf{x}_i, \mathbf{x}_k)$$

- **SEPARATION** of a pair of clusters
  measure how distinct or well-separated a cluster is from other cluster

$$\text{sep}(C_h, C_f) = \sum_{\substack{x_i \in C_h \\ x_k \in C_f}} \text{dist}(\mathbf{x}_i, \mathbf{x}_k)$$

- **OVERALL COHESION**

$$\text{coes}(\mathcal{C}) = \sum_{C_h \in \mathcal{C}} \text{coes}(C_h)$$

- **OVERALL SEPARATION**

$$\text{sep}(\mathcal{C}) = \sum_{\substack{C_h \in \mathcal{C} \\ C_f \in \mathcal{C}}} \text{sep}(C_h, C_f)$$

The lower the cohesion and the higher the separation are, the better the clustering is.

SILHOUETTE COEFFICIENT:

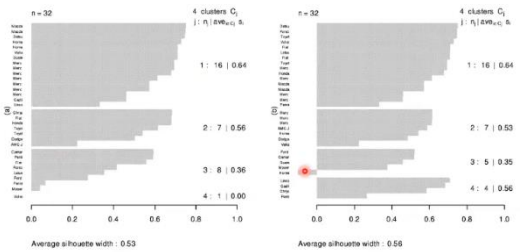
- compute the average distance u_i of $\mathbf{x}_i$ from all the other points in the same cluster
- compute the average distance of $\mathbf{x}_i$ from all the other points comprised in a different cluster
  (let v_i be the minimum of these distances)
- the silhouette coefficient is given by

$$\text{silh}(\mathbf{x}_i) = \frac{v_i - u_i}{\max(u_i, v_i)}$$

- within [-1,1]
- the closer to 1 the better
- average silhouette

You define the silhouette coefficient for each observation.

You consider the average distance of an observation with the points in its cluster and the average distance of the observation with the points of the closest cluster. It is usually positive but not always positive (in case of unnatural assignment).



The second clustering has a higher average silhouette, but one observation has a negative index, it should be re-assigned.

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

6. Association Rules

Find interesting **REGULAR PATTERNS** and **RECURRENCES** within a large set of transactions

Some application domains:

- **market basket analysis**
  find recurrent rules relating the purchase of a group of products to the purchase of other products
- **credit card transactions analysis**
  analyze the purchase via credit cards to direct future promotions (product and services are virtually infinite)
- **web mining**
  understand the patterns of navigation paths
- **fraud detection**
  identify potential fraudulent behavior from a set of transactions consisting of incident reports and applications for compensation

Unsupervised learning tasks. Associate a probability to a rule describing a phenomenon.

Our associative rules are not time dependent, there are also time dependent ones.

Consider a set of n objects (items):

$$\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$$

A **K-ITEMSET** is a generic subset $L \subseteq \mathcal{O}$ containing k objects

A **TRANSACTION** is a generic itemset recorded in a database in a single activity

The data set is composed by a list of m transactions T_i, each associated to a unique identifier t_i

Transaction T contains itemset L if $L \subseteq T$

N is very large, and correspond to all the possible items (usually products). K is very small, just the selected object (usually bought products by a costumer). A transaction is a k-itemset recorded in a single activity (one purchase).

List of transactions

identifier t_i	transaction T_i
001	$\{a, c\}$
002	$\{a, b, d\}$
003	$\{b, d\}$
004	$\{b, d\}$
005	$\{a, b, c\}$
006	$\{b, c\}$
007	$\{a, c\}$
008	$\{a, b, e\}$
009	$\{a, b, c, e\}$
010	$\{a, e\}$

$\mathcal{O} = \{a, b, c, d, e\} = \{\text{bread, milk, cereals, coffee, tea}\}$

The list of m transactions may be expressed by means of a bi-dimensional matrix:

- columns correspond to objects in set $\mathcal{O}$
- rows correspond to transactions
- the generic elements of the matrix is given by $x_{ij} = \begin{cases} 1 & \text{if } o_j \text{ belongs t} T_i \\ 0 & \text{otherwise} \end{cases}$

In a real-world example there are many 0 compared to 1, so the matrix input is not very used.

Matrix of transactions

identifier t_i	a	b	c	d	e
001	1	0	1	0	0
002	1	1	0	1	0
003	0	1	0	1	0
004	0	1	0	1	0
005	1	1	1	0	0
006	0	1	1	0	0
007	1	0	1	0	0
008	1	1	0	0	1
009	1	1	1	0	1
010	1	0	0	0	1

The **EMPIRICAL FREQUENCY** of itemset L is defined as the number of transactions in the data set containing L

When dealing with a large sample, the ratio $f(L)/m$ approximates the probability of occurrence of the itemset L

$\mathcal{O} = \{a, b, c, d, e\} = \{\text{bread, milk, cereals, coffee, tea}\}$

identifier t_i	a	b	c	d	e
001	1	0	1	0	0
002	1	1	0	1	0
003	0	1	0	1	0
004	0	1	0	1	0
005	1	1	1	0	0
006	0	1	1	0	0
007	1	0	1	0	0
008	1	1	0	0	1
009	1	1	1	0	1
010	1	0	0	0	1

$L = \{a, c\} \Rightarrow f(L) = 4 \Rightarrow \Pr(L) \approx 4/10 = 0.4$

Let Y and Z be two propositions which may be true or false

A **RULE** is an implication in the form Y $\Rightarrow$ Z : " if Y is true, then Z is also true "

A rule is called **PROBABILISTIC** if the validity of Z is associated with a certain probability p : " if Y is true, then Z is also true with probability p "

Consider two disjoint itemsets, $L \subset \mathcal{O}$ and $H \subset \mathcal{O}$, and the transaction T.

An **ASSOCIATIVE RULE** is a probabilistic implication denoted as L $\Rightarrow$ H with the following meaning:

" if L is contained in T, then H is also contained in T with probability p "

L (↓ body) and H (↓ head) could be 1-itemsets. So it becomes if this person bough cereals, with a 90% probability also bought milk. You will then put the two objects far away, so that the costumer spend more time in the store, and never do a simultaneous promotion of the two.

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

6.1 Metrics for Assessing Associative Rules

Confidence and support

CONFIDENCE $p = \text{conf}\{L \Rightarrow H\} = \frac{f(L \cup H)}{f(L)}$

- proportion of transactions containing H among those including L (reliability of the rule)
- it approximates the conditional probability that H belongs to T given that L belongs to T

SUPPORT $s = \text{supp}\{L \Rightarrow H\} = \frac{f(L \cup H)}{m}$

- proportion of transactions containing both H and L (frequency of the pair L-H)
- it approximates the probability that L and H are both contained in a future transaction

$\mathcal{O} = \{a, b, c, d, e\}$ = {bread, milk, cereals, coffee, tea}

$L = \{a, c\}$
$H = \{b\}$

identifier t_i	a	b	c	d	e
001	1	0	1	0	0
002	1	1	0	1	0
003	0	1	0	1	0
004	0	1	0	1	0
005	1	1	1	0	0
006	0	1	1	0	0
007	1	0	1	0	0
008	1	1	0	0	1
009	1	1	1	0	1
010	1	0	0	0	1

$p = \text{conf}\{L \Rightarrow H\} = 0.5$ $\quad s = \text{supp}\{L \Rightarrow H\} = 0.2$

Machine Learning © Carlo Vercellis POLITECNICO MILANO 1863 49

Confidence and Support are kind of hyperparameter. Very hard to identify the correct value, you try high and iteratively lower it.

Strong association rules

Objective ⇒ find the **STRONG** association rules:

$s \geq s_{min} \qquad p \geq p_{min}$

The number of possible rules increases exponentially as the number of objects increases ...

⇓

... a two-phase procedure is applied:

- generation of **FREQUENT ITEMSETS**
- generation of strong rules

From a set of 20 objects it is possible to generate about 3.5 billions of association rules!

Machine Learning © Carlo Vercellis POLITECNICO MILANO 1863 49

Strong rules are the relevant one.

But not always strong rules are actually relevant:

Strong association rules

Strong rules are always meaningful?

A retailer whishes to analyze a set of transactions to identify associations between sales of color printers and sales of digital cameras:

- 1000 transactions
- 600 include digital cameras
- 750 include color printers
- 400 include both products

probability of purchasing a printer conditioned to the purchase of a camera

The rule { digital camera } ⇒ { color printer } has support 0.4 and confidence 0.66

What is the probability of purchasing a printer?...0.75, which is greater than 0.66!

printers and cameras show a negative correlation... ...the purchase of a camera reduces the probability of buying a printer!

Machine Learning © Carlo Vercellis POLITECNICO MILANO 1863 49

6.1.1 Lift Index

A third measure of significance is the **LIFT** :

$$l = \text{lift}\{L \Rightarrow H\} = \frac{\text{conf}\{L \Rightarrow H\}}{f(H)} = \frac{f(L \cup H)}{f(L)f(H)}$$

- the lift is greater than 1: body and head are *positively associated*
  the rule is effective in predicting the presence of the head in a given transaction
- the lift is lower than 1: body and head are *negatively associated*
  the rule is less effective than the estimate obtained by the frequency of the head

For the former example:

$f(L \cup H)$ = 400/1000 = 0.4
$f(L)$ = 600/1000 = 0.6 $\qquad \text{lift}\{L \Rightarrow H\}$ = 0.4/0.45 ≈ 0.89
$f(H)$ = 750/1000 = 0.75

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

6.2 Solution Algorithm

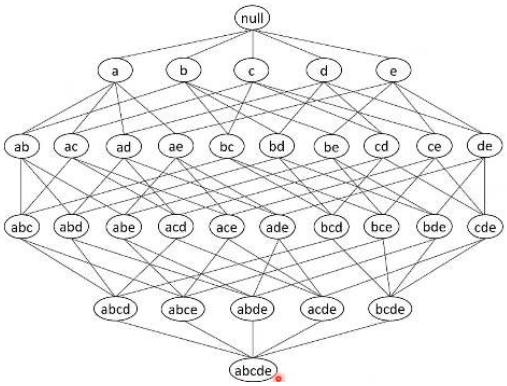
The support of the rule L ⇒ H only depends on the set $L \cup H$ given by the union of the itemsets L and H

⇓

If the itemset $L \cup H$ is not frequent, we can exclude from the analysis all the rules obtained by using all the proper subsets $L \cup H$ ...

$$\{a,b\} \Rightarrow \{c\} \qquad \{a,c\} \Rightarrow \{b\}$$
$$\{b,c\} \Rightarrow \{a\} \qquad \{a\} \Rightarrow \{b,c\}$$
$$\{b\} \Rightarrow \{a,c\} \qquad \{c\} \Rightarrow \{a,b\}$$

... the real problem, therefore, is how to determine the frequent itemsets !



Through the itemset grid, we can identify frequent itemset.

A data set of *m* transactions defined over a set of *n* objects may contain up to 2^n-1 frequent itemsets (excluding the empty set) ...exhaustive enumeration is impracticable...

The APRIORI ALGORITHM is an effective method for extracting strong rules.
It relies on the following property (APRIORI PRINCIPLE):

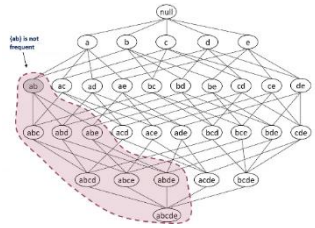
"if an itemset is frequent, then all its subsets are also frequent"

⇓

if an itemset is not frequent, then each of the itemset containing it must turn out to be not frequent!

Once a non-frequent itemset is identified in the course of the algorithm, all the other itemsets (with greater cardinality) containing it are implicitly eliminated and excluded from the analysis

You start from the top of the itemset and if one itemset is not frequent all the lower ones linked to it are not frequent.



The Apriori algorithm generates the frequent itemsets iteratively, starting from the frequent 1 – itemset to the frequent k – itemsets

- Phase 1 - initialization
  Compute the relative frequency of each object.
  Discard the objects with a frequency smaller than the given minimum threshold.

$\mathcal{O} = \{a, b, c, d, e\}$ = {bread, milk, cereals, coffee, tea}

itemset	relative frequency	status
$\{a\}$	$7/10 = 0.7$	frequent
$\{b\}$	$7/10 = 0.7$	frequent
$\{c\}$	$5/10 = 0.5$	frequent
$\{d\}$	$3/10 = 0.3$	frequent
$\{e\}$	$3/10 = 0.3$	frequent

- Phase 1 - iteration 1
  The candidate 2 – itemsets are obtained from the 1 – itemset found during the former iteration.
  Non-frequent itemsets are discarded.

$\mathcal{O} = \{a, b, c, d, e\}$ = {bread, milk, cereals, coffee, tea}

itemset	relative frequency	status
$\{a, b\}$	$4/10 = 0.4$	frequent
$\{a, c\}$	$4/10 = 0.4$	frequent
$\{a, d\}$	$1/10 = 0.1$	not frequent
$\{a, e\}$	$3/10 = 0.3$	frequent
$\{b, c\}$	$3/10 = 0.3$	frequent
$\{b, d\}$	$3/10 = 0.3$	frequent
$\{b, e\}$	$2/10 = 0.2$	frequent
$\{c, d\}$	$0/10 = 0.0$	not frequent
$\{c, e\}$	$1/10 = 0.1$	not frequent
$\{d, e\}$	$0/10 = 0.0$	not frequent

- Phase 1 - iteration 2
  The candidate 3 – itemsets are obtained from the 2 – itemsets found during the former iteration.
  Non-frequent itemsets are discarded.

$\mathcal{O} = \{a, b, c, d, e\}$ = {bread, milk, cereals, coffee, tea}

itemset	relative frequency	status
$\{a, b, c\}$	$2/10 = 0.2$	frequent
$\{a, b, e\}$	$2/10 = 0.2$	frequent

The first phase of the algorithm terminates since no candidate 4 – itemsets can be generated!

Strong rules generation

itemset	rule	confidence	status
$\{a, b\}$	$\{a \Rightarrow b\}$	$\rho = 4/7 = 0.57$	strong
$\{a, b\}$	$\{b \Rightarrow a\}$	$\rho = 4/7 = 0.57$	strong
$\{a, c\}$	$\{a \Rightarrow c\}$	$\rho = 4/7 = 0.57$	strong
$\{a, c\}$	$\{c \Rightarrow a\}$	$\rho = 4/5 = 0.80$	strong
$\{a, e\}$	$\{a \Rightarrow e\}$	$\rho = 3/7 = 0.43$	not strong
$\{a, e\}$	$\{e \Rightarrow a\}$	$\rho = 3/3 = 1.00$	strong
$\{b, c\}$	$\{b \Rightarrow c\}$	$\rho = 3/7 = 0.43$	not strong
$\{b, c\}$	$\{c \Rightarrow b\}$	$\rho = 3/5 = 0.60$	strong
$\{b, d\}$	$\{b \Rightarrow d\}$	$\rho = 3/7 = 0.43$	not strong
$\{b, d\}$	$\{d \Rightarrow b\}$	$\rho = 3/3 = 1.00$	strong
$\{b, e\}$	$\{b \Rightarrow e\}$	$\rho = 2/7 = 0.29$	not strong
$\{b, e\}$	$\{e \Rightarrow b\}$	$\rho = 2/3 = 0.67$	strong
$\{a, b, c\}$	$\{a, b \Rightarrow c\}$	$\rho = 2/4 = 0.50$	not strong
$\{a, b, c\}$	$\{c \Rightarrow a, b\}$	$\rho = 2/5 = 0.40$	not strong
$\{a, b, c\}$	$\{a, c \Rightarrow b\}$	$\rho = 2/4 = 0.50$	not strong
$\{a, b, c\}$	$\{b \Rightarrow a, c\}$	$\rho = 2/7 = 0.29$	not strong
$\{a, b, c\}$	$\{b, c \Rightarrow a\}$	$\rho = 2/3 = 0.67$	strong
$\{a, b, c\}$	$\{a \Rightarrow b, c\}$	$\rho = 2/7 = 0.29$	not strong
$\{a, b, e\}$	$\{a, b \Rightarrow e\}$	$\rho = 2/4 = 0.50$	not strong
$\{a, b, e\}$	$\{e \Rightarrow a, b\}$	$\rho = 2/3 = 0.67$	strong
$\{a, b, e\}$	$\{a, e \Rightarrow b\}$	$\rho = 2/3 = 0.67$	strong
$\{a, b, e\}$	$\{b \Rightarrow a, e\}$	$\rho = 2/7 = 0.29$	not strong
$\{a, b, e\}$	$\{b, e \Rightarrow a\}$	$\rho = 2/2 = 1.00$	strong
$\{a, b, e\}$	$\{a \Rightarrow b, e\}$	$\rho = 2/7 = 0.29$	not strong

- Phase 2 - step 1
  Split each frequent itemset B into two disjoint non-empty subsets L and H = B – L , according to each possible combination
  Generate all the corresponding candidate rules ⟶ given a frequent k-itemesets it is possible to generate 2^k-2 candidate rules which may by strong rules (excluding those with empty body or head)
- Phase 2 - step 2
  For each candidate rule L ⇒ H compute the confidence
  ↓
  very fast step – the frequencies of (L∪H) and L are known at the end of the first phase
- Phase 2 - step 3
  Include into the list of strong rules all the rules provided with a confidence greater than the prefixed minimum threshold

This document is available on studocu

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)

The computational effort required grows exponentially as the number *n* of objects increases

To improve the efficiency:

- resort to advanced data structures (dictionaries, binary trees)
- split the data set into disjoint subsets of transactions and apply the algorithm to each subset (local frequent itemsets)
- use the algorithm on a given sample of transactions
- resort to hierarchies of objects
- discard less interesting objects in the preprocessing phase

The aforementioned rules are *binary*, *asymmetric* and *one-dimensional* rules

↓ presence vs. absence ↓ presence is more relevant

More general association rules:

- rules for binary symmetric variables ({male,female}, {yes,no}, ...)
  → two asymmetric binary variables are introduced for each symmetric variable
- rules for categorical variables (education, profession,...)
  → a set of asymmetric binary variables is introduced for each categorical variable
- rules for continuous variables (age, income,...)
  → discretization is first applied to obtain a categorical variable
- multi-dimensional rules
- sequential rules (when transactions are recorded according to a specific temporal sequence)

Scaricato da Amirhossein Jandaghian (amirh.jandaghian@gmail.com)