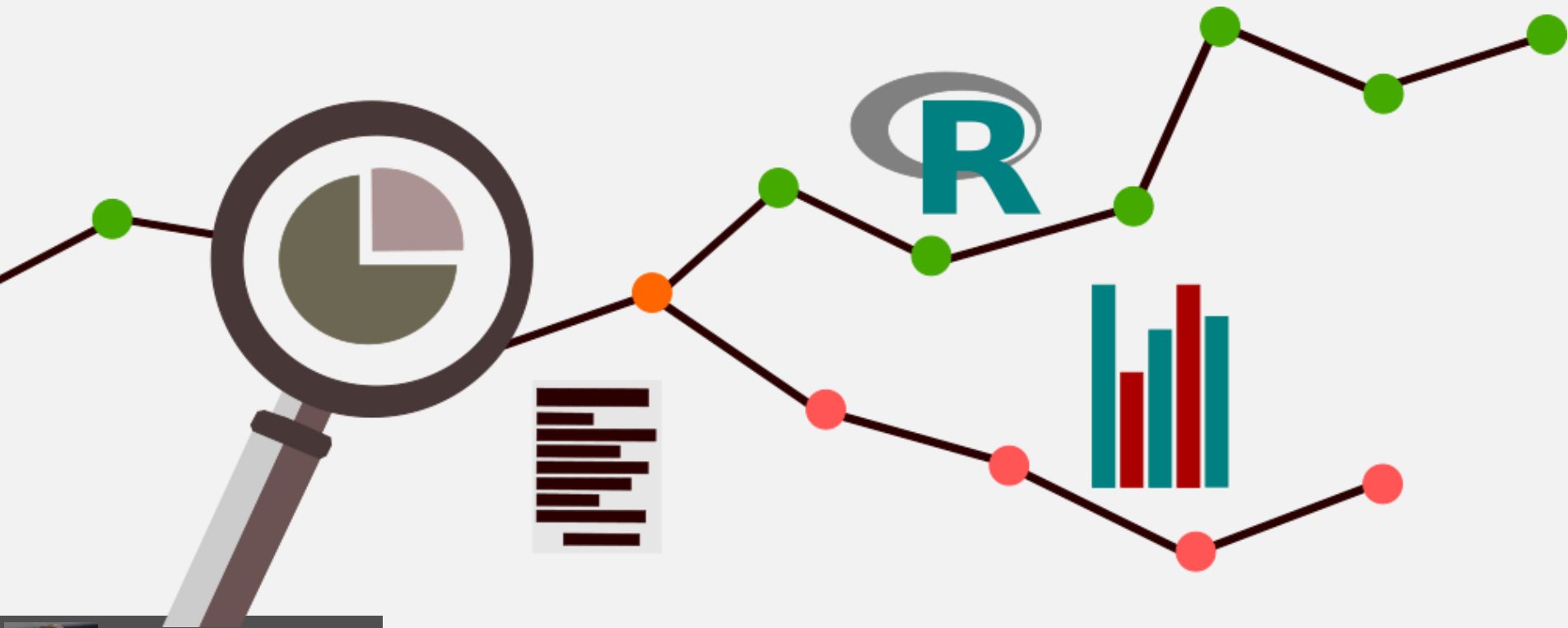


Decision Support Systems and Business Analytics with R

Section 5: Business Data Analytics



Dominik Jung
d.jung@kit.edu

5

Business Data Analytics

5.1 Introduction into Business Data Analytics

5.2 Modelling and Analytics

5.3 Bias, Variance, and Model Performance

5.4 Clustering

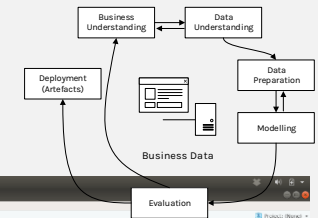
5.5 Association Analysis

5.6 Classification

5.7 Regression

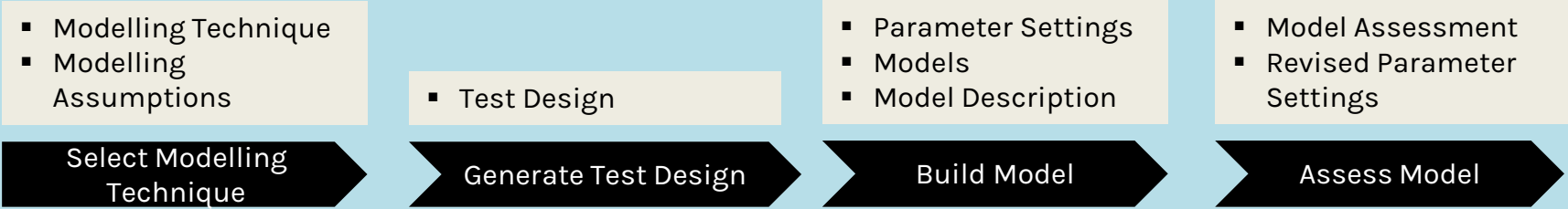
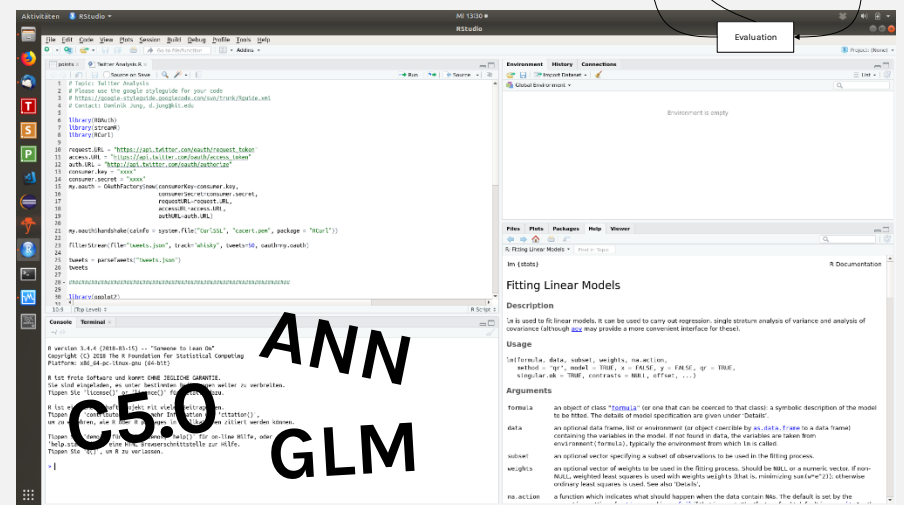
5.8 Time-Series

Business Data Analytics



► What activities are related to the modelling phase?

- Modeling techniques selection and application
- Parameter calibration and model assessment
- **Methods:** Regression, classification, association analysis, clustering, time-series forecasting



1. Select Modelling Technique

- Modelling Technique
- Modelling Assumptions

- Select technique
- Identify any built-in assumptions made by the technique about the data
 - **Example:** *quality, format, distribution*
- Compare these assumptions with those in the Data Description Report and make sure that these assumptions hold.
- Preparation Phase if necessary.



1. Select modelling technique: Modelling technique

- Record an actually used modeling technique

Deliverables

- appropriate technique chosen according to the tool selected

Select Modelling
Technique

Generate Test Design

Build Model

Assess Model

1. Select modelling technique: Modelling assumptions

- **Record specific assumptions for an actually used modeling technique..**

Deliverables

- assumptions about the data for an actually used modeling technique and their comparison with the Data description report (task II.2.)

Select Modelling
Technique

Generate Test Design

Build Model

Assess Model

2. Generate Test Design

- Test Design

- Describe the intended plan for train, test and evaluate the models.
- How to divide the dataset into training, test and validation sets.
- Decide on necessary steps (number of iterations, number of folds etc.).
- Prepare data required for test.



2. Generate Test Design: Test design

- **Describe plans for training, testing and evaluating the model.**

Deliverables

- existing test designs for the data mining goals
- necessary steps (iterations, folds, ...)
- prepared test data

Select Modelling
Technique

Generate Test Design

Build Model

Assess Model

3. Build Model

- Parameter Settings
- Models
- Model Description

- Set initial parameters and document reasons for choosing those values.
- Run the selected technique on the input dataset. Post-process data mining results
 - **Example:** editing rules, display trees
- Record parameter settings used to produce the model
- Describe the model, its special features, behavior and interpretation



3. Build Model: Parameter setting

- **Set initial parameters and document reasons for the chosen values.**

Deliverables

Select Modelling
Technique

Generate Test Design

Build Model

Assess Model

3. Build Model: Models

- **Run the selected technique and post-process the results.**

Deliverables

Select Modelling
Technique

Generate Test Design

Build Model

Assess Model

3. Build Model: Model description

- **Describe the resulting model and assessment of its properties.**

Deliverables

- characteristics of the model and its parameter settings
- detailed description of the model with technical information
- the interpretation of the model
- conclusions regarding possible patterns in data

Select Modelling
Technique

Generate Test Design

Build Model

Assess Model

4. Assess Model

- Model Assessment
- Revised Parameter Settings

- Evaluate result with respect to evaluation criteria. Rank results with respect to success and evaluation criteria and select best models.
- Interpret results in business terms. Get comments by domain experts.
- Check plausibility of model.
- Check model against given knowledge base. Is the discovered information **novel** and **useful**?
- Check result reliability. Analyze potentials for deployment of each result.



4. Assess Model: Model assessment

- **Summarize the results of this task.**

Deliverables

- test results of models acquired, their comparison and interpretation
- best models selected and their interpretation in business terms
- comments from domain experts on reliability, plausability, usefulness and novelty as well as impacts of these models
- analysis of potentials of deployment of these models
- insights why a certain modeling technique leads to good/bad results

Select Modelling
Technique

Generate Test Design

Build Model

Assess Model

4. Assess Model: Revised parameter settings

- **Adjust parameters to lead to better results.**

Deliverables

Select Modelling
Technique

Generate Test Design

Build Model

Assess Model

1.1 Modelling and Analytics

- Use cases in analytics
- Popular modelling techniques

Most common business analytics jobs

Problem	Business Perspective	Techniques
Find Clusters/Outliers	<ul style="list-style-type: none">▪ Are there different types of users▪ Can we put different products together into distinct/different groups?	<ul style="list-style-type: none">▪ Clustering▪ Outlier-Analysis
Find Relationships	<ul style="list-style-type: none">▪ If a customer buys product A, what does he buy next?▪ Which product sets belong together?	<ul style="list-style-type: none">▪ Association Analysis
Predict Classes	<ul style="list-style-type: none">▪ Is this customer solvent or not?▪ Will this customer send back this shipping or not?	<ul style="list-style-type: none">▪ Decision Trees▪ Logistic Regression▪ Support-Vector Machines
Predict Values	<ul style="list-style-type: none">▪ Does a new label increase the?▪ Is there a relationship between Sales and Commercials?	<ul style="list-style-type: none">▪ Regression▪ Support-Vector Machines
Predict Developments	<ul style="list-style-type: none">▪ How will the value of our products develop?▪ What future developments are likely?	<ul style="list-style-type: none">▪ Time-Series Forecasting

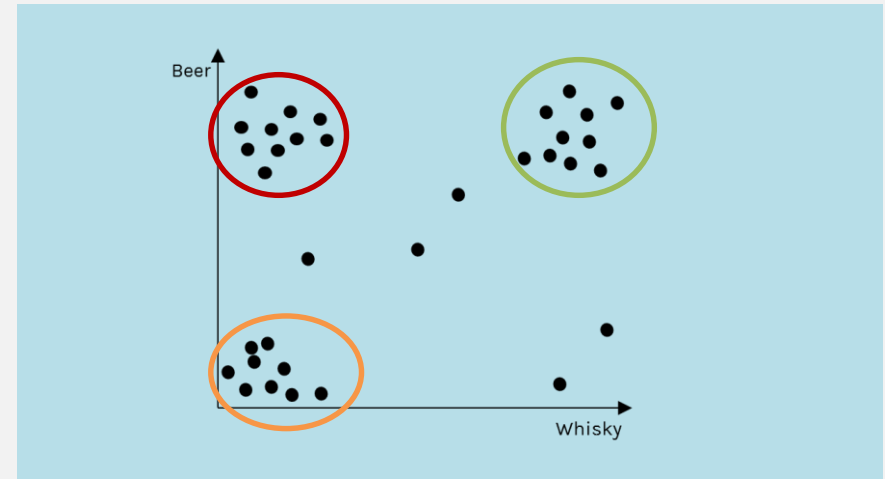


There will be further advanced techniques we will discuss in „Advanced Analytics with R“, another lecture covering topics like e.g. ANN, FP Growth techniques, Expectation Maximization etc.

Clustering

► **Forming groups of objects in a way that the same group (clusters) are more similar to each other than to those in other groups**

- Clustering is an unsupervised analytics technique
- Data structuring tool generally used for exploratory rather than confirmatory analysis
- Techniques differ significantly in their understanding of what constitutes a cluster



Use Cases:

- **Clustering users:** Finding users with similar interests and preferences for recommendations, e.g. Netflix or Amazon
- **Data Stream Clustering:** Cluster different telephone records, multimedia data, financial transactions, user log-files etc. for further analysis

Association Analysis

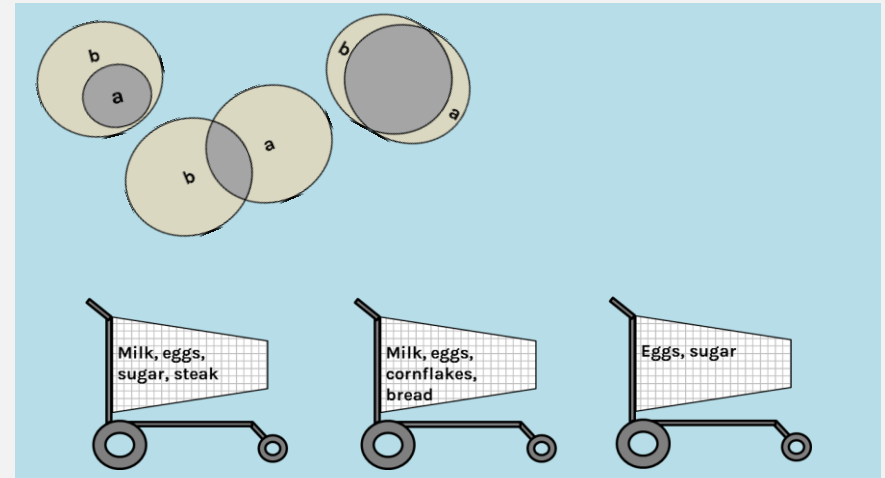
► Describes how specific phenomena relate to each other.

- Used to discover interesting regularities between variables
- The uncovered relationships can be represented in form of association rules
- does not consider the order of items either within a transaction or across transactions.

Example:

$\{\text{Whisky, Potatoes}\} \rightarrow \{\text{Steak}\}$

Customers buying whisky and potatoes together are also likely to buy a steak.



Use Cases:

- **Shopping basket analysis:** Identify customer preferences by finding associations and correlations between different products that customers bought together.
- **(Web) Pattern Mining:** use multi-dimensional association rules to identify behavioral patterns based on log files.

Classification

➤ Classify examples into given set of categories

- Technique to categorize data into a desired and distinct number of classes where we can assign label to each class
- The distinct rules can be represented in form of hierarchical models like e.g. decision trees

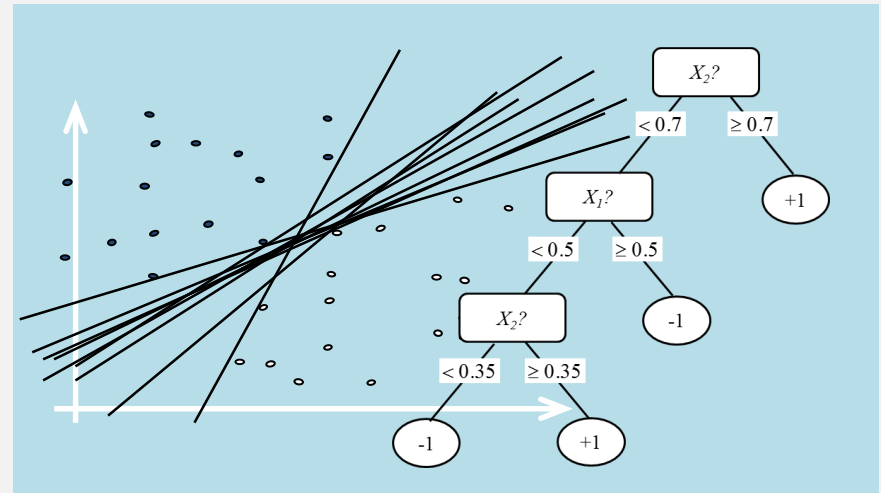
Example:

Age < 25

Income > 3200 Euro

True

Consumer older than 25 years with an income of at least 3200 Euro are creditworthy.



Use Cases:

- **Predict Customer loyalty:** Identify customer preferences by finding associations and correlations between different products that customers bought together.
- **Spam Detection:** use classification techniques to separate spam emails and non spam email based on old files.

Regression Analysis

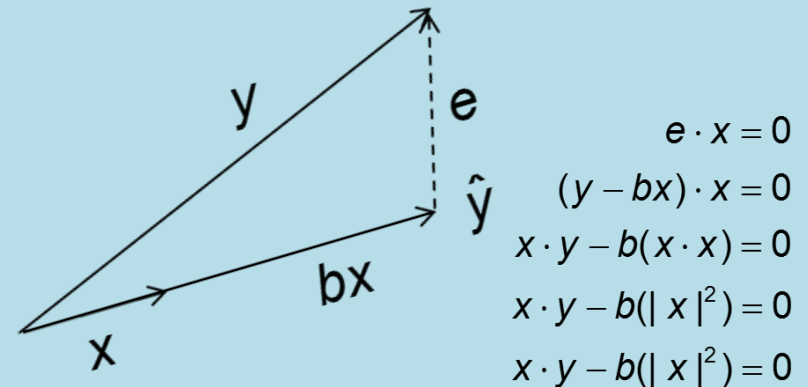
➤ Describes how specific phenomena relate to each other.

- Used to predict the value of one variable based on the value of other variables
- Independenten variables can be affected by each other but it does not mean that this dependency is both ways as is the case with correlatin analysis

Example:

$\text{Whisky_Consume} = X1 \cdot \text{Age} + X2 \cdot \text{Income} + e$

How much whisky someone drinks depends on age and income



Use Cases:

- **Consumer Analysis:** Levels of customer satisfaction affects customer loyalty
- **Pricing:** How neighbourhood and size affect the listing price of houses
- **Matchmaking:** Find the love of your life via online dating ;-)

Recommended literature

Wu X et al. (2008)

Knowl Inf Syst (2008) 14:1–37
DOI 10.1007/s10115-007-0114-2

SURVEY PAPER

Top 10 algorithms in data mining

Xindong Wu · Vipin Kumar · J. Ross Quinlan · Joydeep Ghosh · Qiang Yang · Hiroshi Motoda · Geoffrey J. McLachlan · Angus Ng · Bing Liu · Philip S. Yu · Zhi-Hua Zhou · Michael Steinbach · David J. Hand · Dan Steinberg

Received: 9 July 2007 / Revised: 28 September 2007 / Accepted: 8 October 2007
Published online: 4 December 2007
© Springer-Verlag London Limited 2007

Abstract This paper presents the top 10 data mining algorithms identified by the IEEE International Conference on Data Mining (ICDM) in December 2006: C4.5, *k*-Means, SVM, Apriori, EM, PageRank, AdaBoost, *k*NN, Naive Bayes, and CART. These top 10 algorithms are among the most influential data mining algorithms in the research community. With each algorithm, we provide a description of the algorithm, discuss the impact of the algorithm, and review current and further research on the algorithm. These 10 algorithms cover classification,



<https://link.springer.com/article/10.1007/s10115-007-0114-2>

1.2 Bias, Variance and Model Performance

- Bias and variance dilemma
- Estimate model performance
- Data splitting for model evaluation
- Model complexity

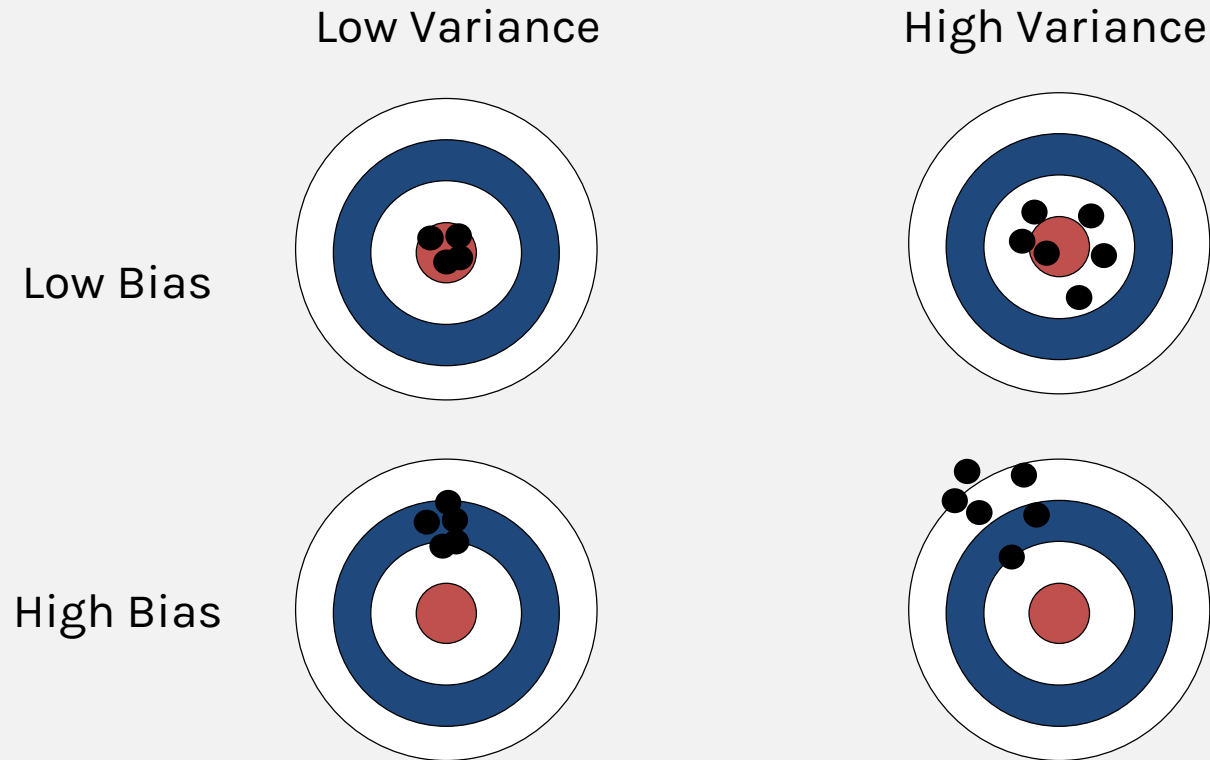
Bias and Variance

- Analytics in a nutshell: Build a statistical model to make predictions about your dataset (the reality) with as few **errors** as possible
- The prediction errors of your final model can be decomposed into two main subcomponents:
 - **Error due to Bias:** difference between the expected prediction and the true value
 - **Error due to Variance:** variability of a model prediction for a given data point



What do you think. What is the relationship between error, bias and variance?

Visualization of bias and variance

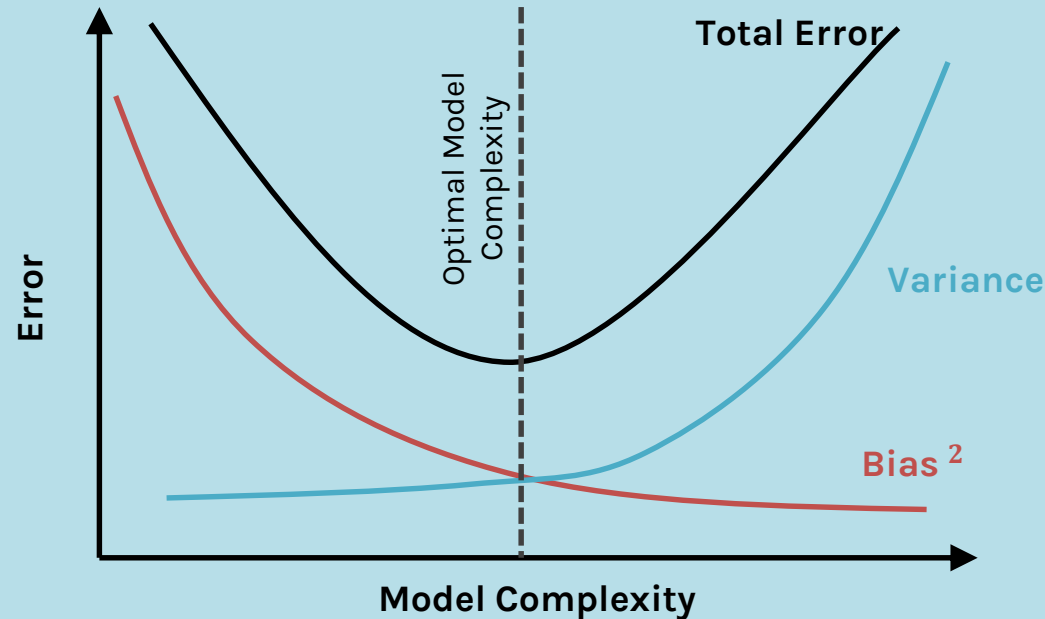


Fitting dilemma

- We have seen bias and variance influence each other: the bias decreases with the complexity of the model, while the variance increases with the complexity of the model
- As a consequence, there are two opposite effects:
 - **Underfitting:** When the model has high bias and low variance, i.e. is too general (high total error)
 - **Overfitting:** When the model has low bias and high variance, i.e. is too specific (high total error)

The bias-variance trade-off

- We need to make a trade-off between “too specific” and “too general”



Noise in sampling

- The error, we discussed on the previous slide deals with errors based on labelled observations. However, the precise value of our labels are often unknown.
- Real-world data is usually noisy, the “real” value is often influenced by its generation or measurement:

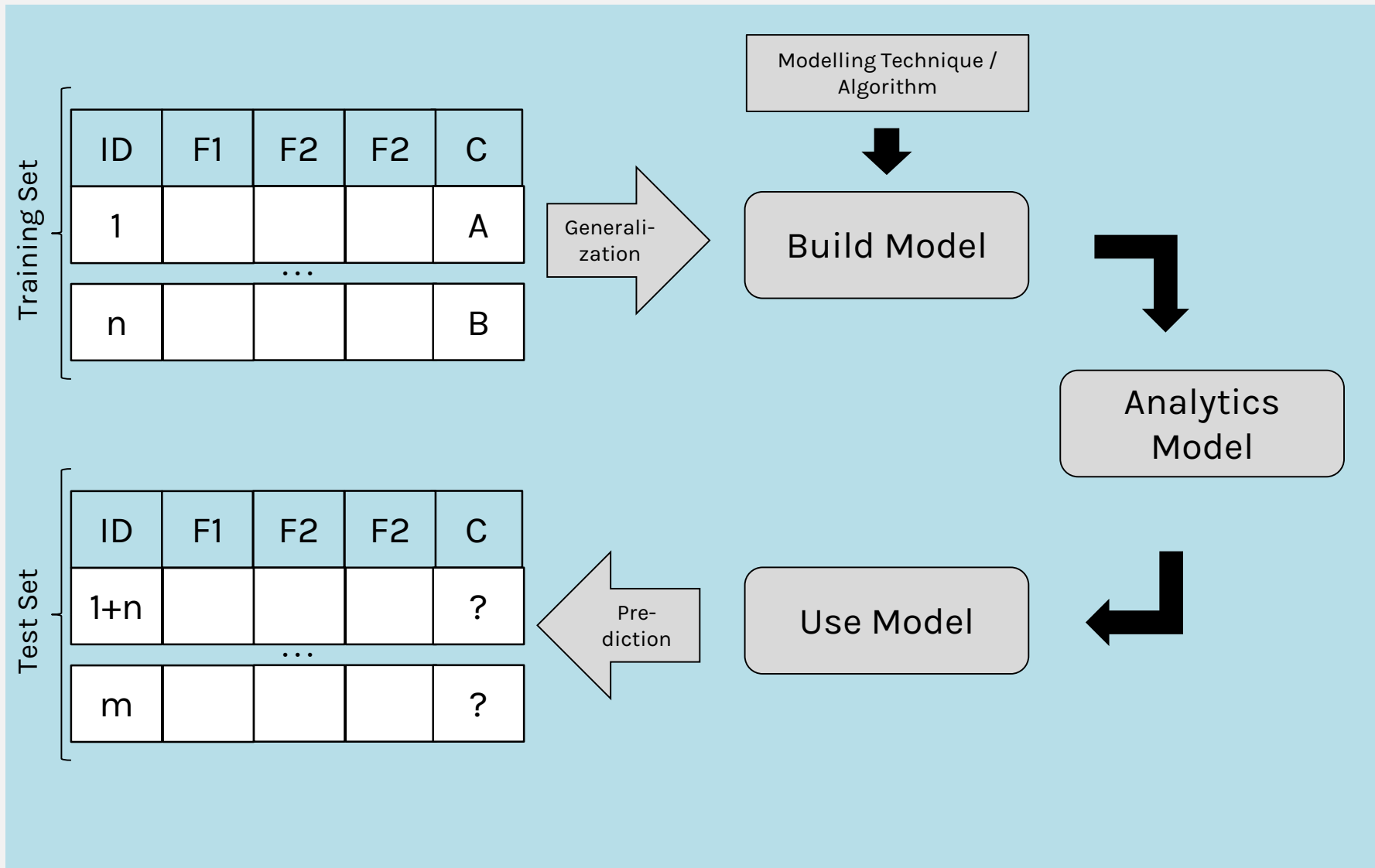
$$y = l(x) + \varepsilon$$



In this phase, we do not care where the noise ε comes from

- There can be more observations with same attribute values but different labels or the other way round.

A simple approach to measure model's performance



Model accuracy and error measures

Accuracy

Model Accuracy: Measures how far off the predicted value is from the actual known value

Loss Function

Observation: y_t
Prediction: \hat{y}_t

Error: $e_t = y_t - \hat{y}_t$

Absolute Error: $|y_t - \hat{y}_t|$

Squared Error: $(y_t - \hat{y}_t)^2$

Accuracy Measures

Measure	Formula
Mean Absolute Error	$MAE = average(e_t)$
Mean Squared Error	$MSE = average(e_t^2)$
Mean Absolute Percentage Error	$MAPE = 100 \cdot average\left(\left \frac{e_t}{y_t}\right \right)$
Mean Absolute Scaled Error	$MASE = \frac{MAE}{Q}$

Q: Scaling constant

Accuracy of classification models

In classification problems, the primary source for accuracy estimation is the **confusion matrix**

		True Class	
		Positive	Negative
Predicted Class	Positive	True Positive Count (TP)	False Positive Count (FP)
	Negative	False Negative Count (FN)	True Negative Count (TN)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

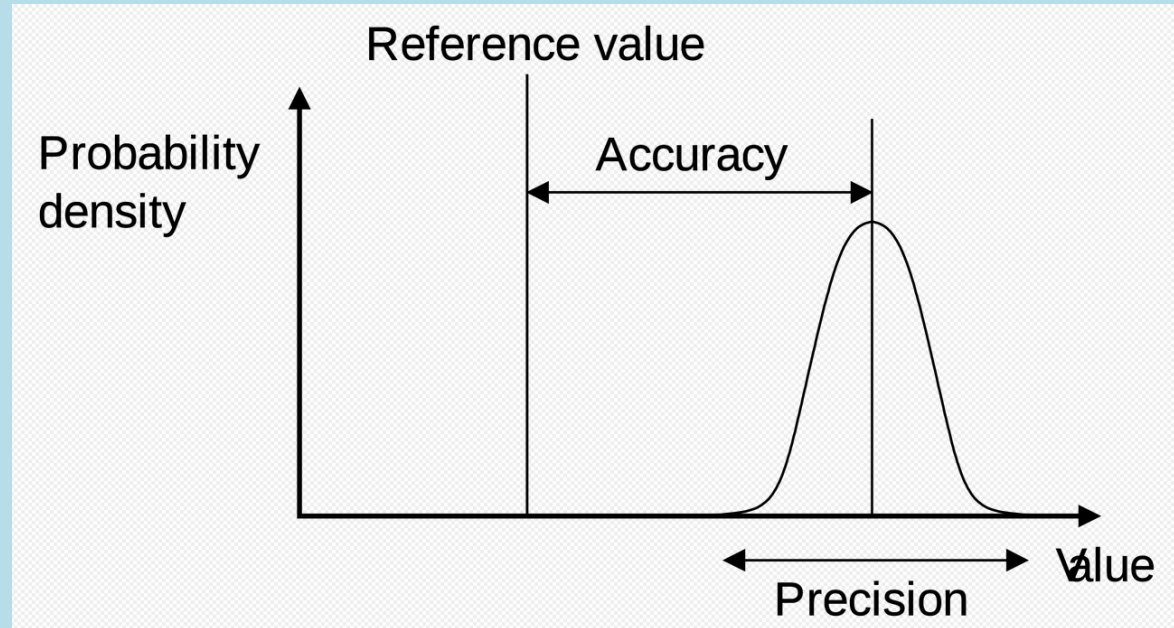
$$True\ Positive\ Rate = \frac{TP}{TP + FN}$$

$$True\ Negative\ Rate = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

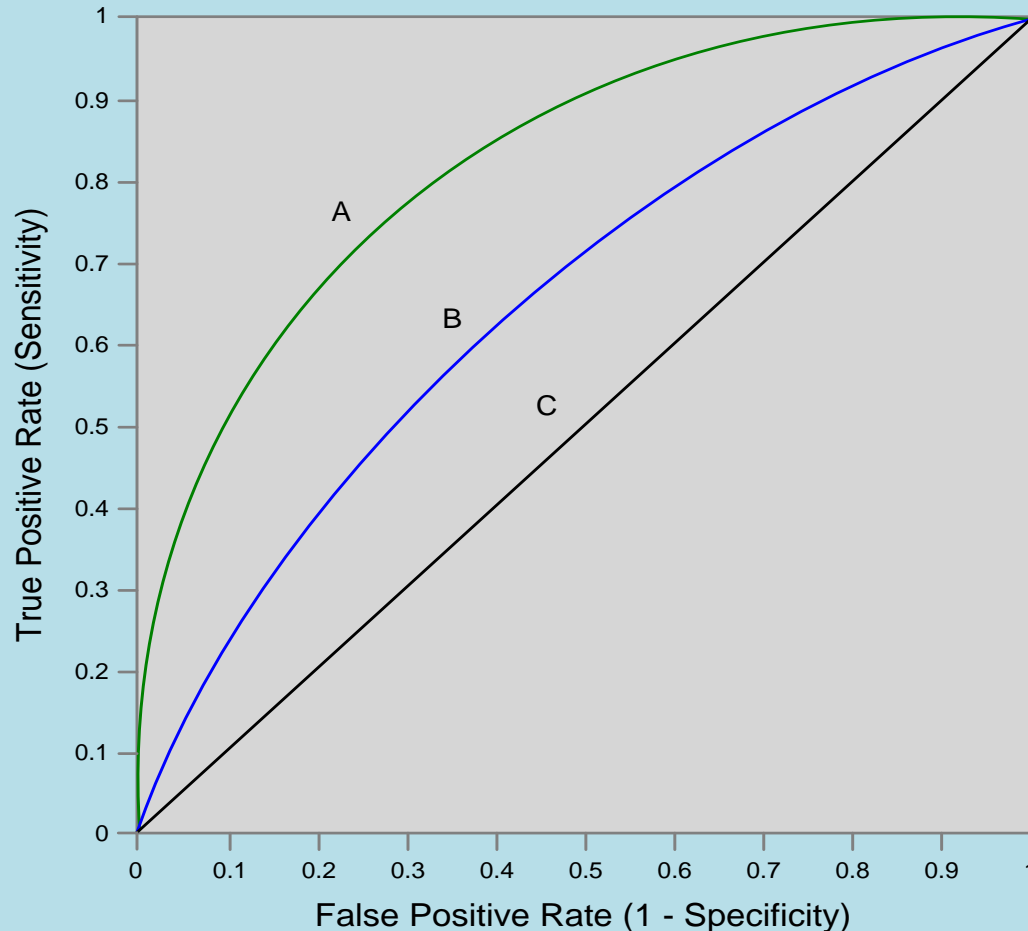
$$Recall = \frac{TP}{TP + FN}$$

Precision and accuracy



- In analytics: bias and variability instead of accuracy and precision
- bias is the amount of inaccuracy and variability is the amount of imprecision.

Visualization of the accuracy – ROC Curve



ROC Curve

```
devtools::install_github(  
  "sachsmc/plotROC")
```

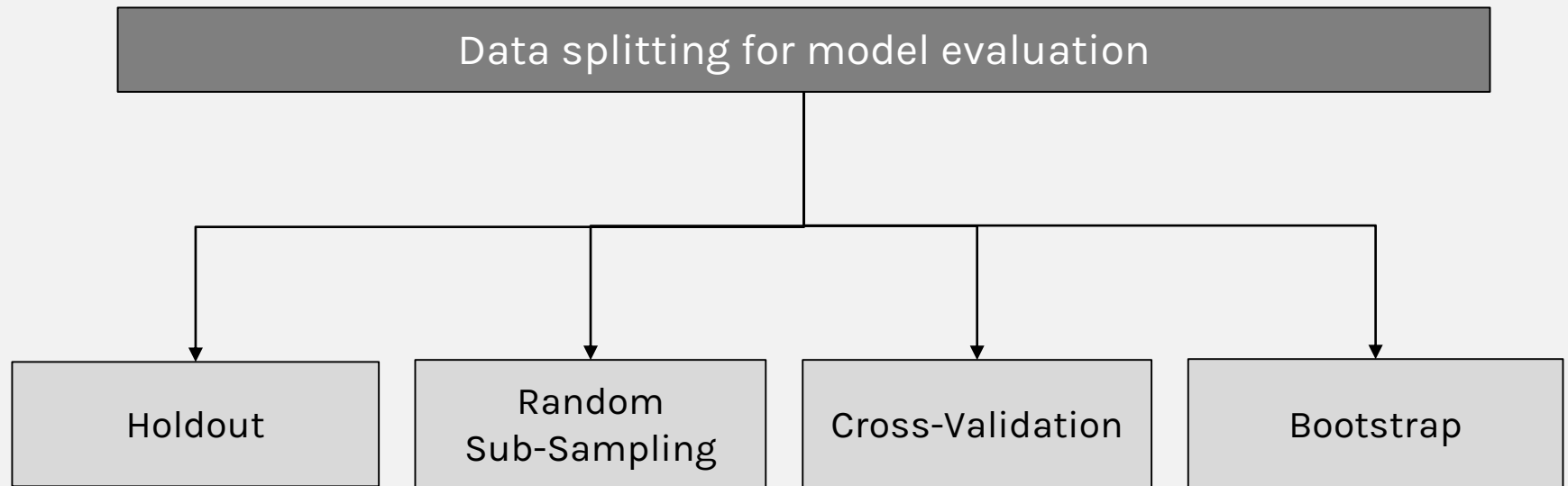
```
ggplot(data,) +  
  geom_roc()
```

- Receiver Operating Characteristic (ROC)
- Used to assess the accuracy of a continuous measurement for predicting a binary outcome



<https://cran.r-project.org/web/packages/plotROC/index.html>

Measure the goodness of a model's prediction

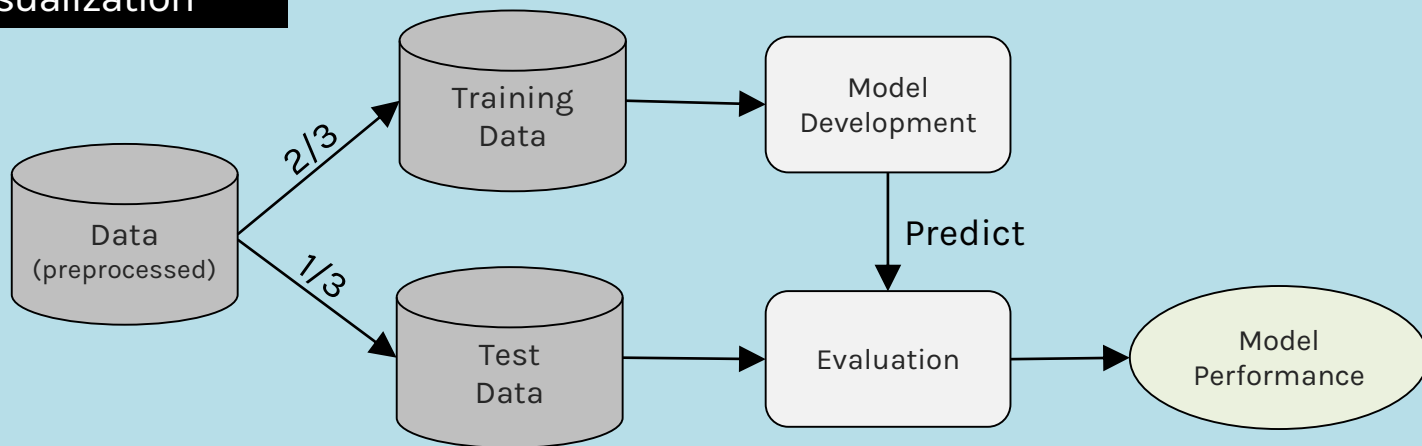


Holdout-Method

Procedure

- Split your labelled data into train and test set, Build your model based on train set, and measure the performance based on the test set
- In problems where we have a sparse dataset we may not be able to afford the “luxury” of setting aside a portion of the dataset for testing
- Since it is a single train-and-test experiment, the holdout estimate of error rate will be misleading if we happen to get an “unfortunate” split

Visualization

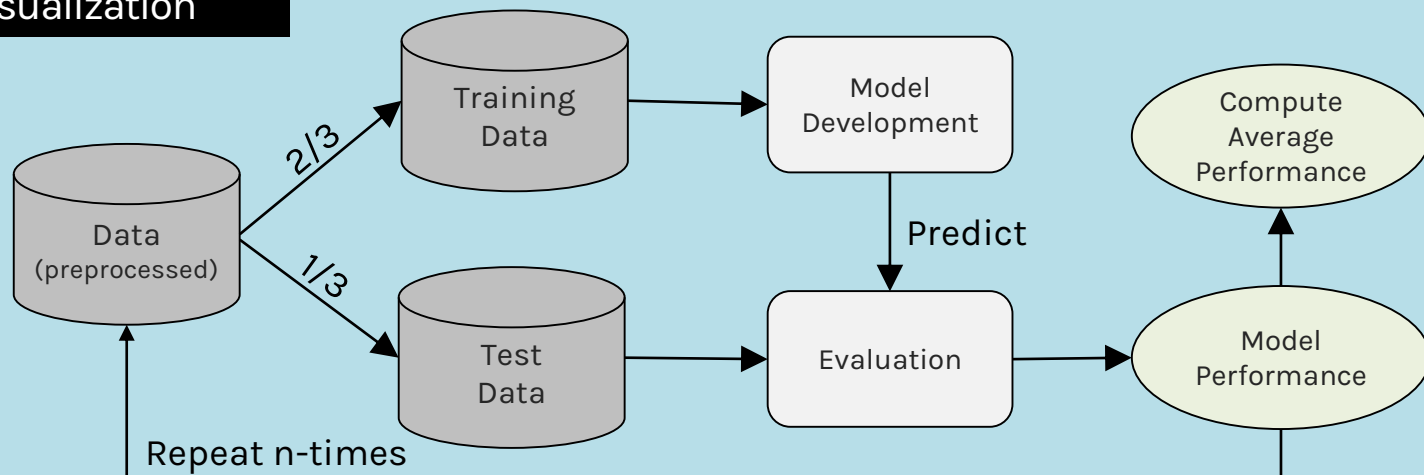


Random Sub-Sampling

Procedure

- Repeated holdout with different samples, but measure the average performance
- Multiple models, where you can choose from
- Same disadvantages then simple holdout
- No control how often an observation is used for model building

Visualization

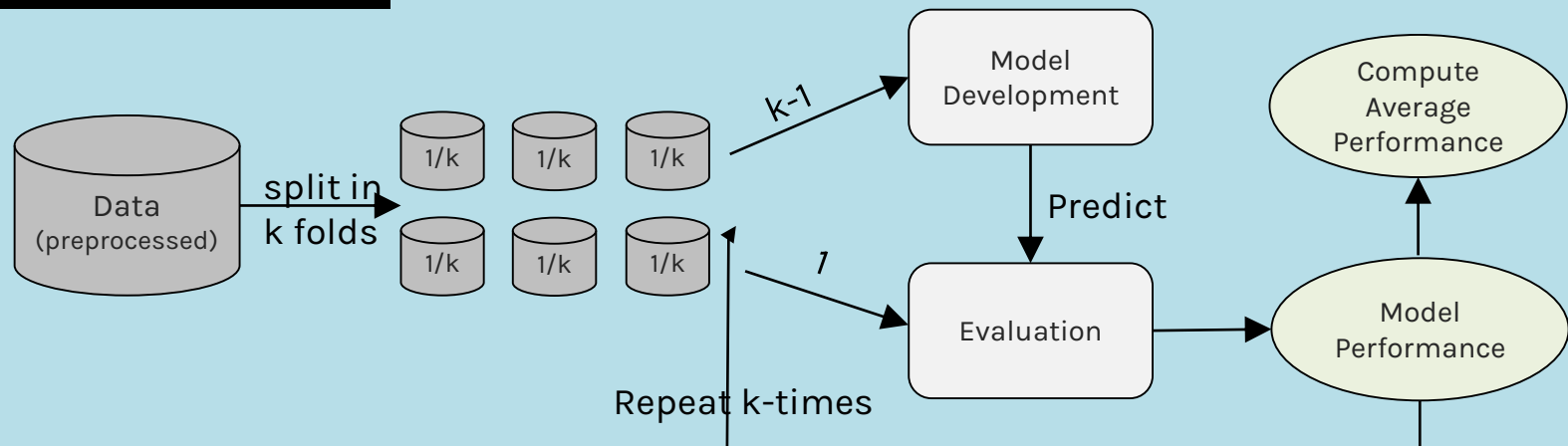


K-fold Cross-Validation

Procedure

- Split data into k same-sized samples, use $k-1$ samples for training, and 1 for testing. Each observation is used the same time for training
- Benefit is that it uses as many model building examples as possible and test sets disjunct
- High complexity of the process (k -runs), reliability of the performance statement is weakened, since these statements are derived from only one example.

Visualization



Bootstrap

- In the previous procedures, an example was considered several times as a training example (in the same cycle).
- Here a training set is generated by random selection from the entire set of classified examples (Sampling with replacement).
- I.e. an observation can occur several times in the training set with a certain probability.
- Probability to be chosen is $1 - (1 - 1/N)^N$
- for N against infinity this value converges to $1 - 1/e = 0,632$

Model complexity

- We talked a lot about errors and increasing accuracy adjusting your model based on different performance measurement and sampling techniques, but there is also an other side...
- **Problem:** What do you do when your model your model starts overfitting?
- **Solution:** Consider the model complexity in your evaluation of the model

Occam's Razor

- If you have two models with the same generalization error, the simpler model (with fewer nodes, elements, predictors, etc.) is preferable.

Pessimistic Error Rate $e_g(T)$

- **Example:** Decision tree
- To the sum of all misclassifications $e(t_i)$ at the leaf nodes above the training data one adds a malus ("penalty") $\Omega(t_i)$ for each leaf node t_i in the tree and refers the result to the number of observations in the training data.

$$e_g(T) = \frac{\sum_{j=1}^k [e(t_i) + \Omega(t_i)]}{\sum_{i=1}^k n(t_i)} = \frac{e(T) + \Omega(T)}{N_i}$$

Minimum Description Length Principle

- For each misclassification a measure is added to the binary coding to punish the complexity of the model.

$$\textit{cost}(\textit{fit}, \textit{data}) = \textit{cost}(\textit{data}|\textit{fit}) + \textit{cost}(\textit{fit})$$

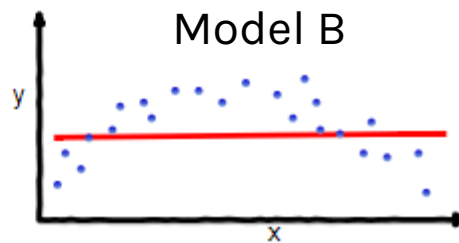
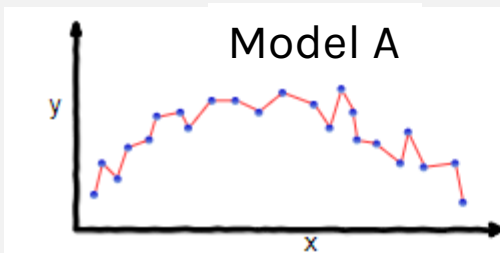
Example: for 16 observations (4 bits) and 3 errors, $\textit{cost}(\textit{data} | \textit{fit}) = 3 \cdot 4 = 12$

- Mostly used for decision trees

Your turn!

Task

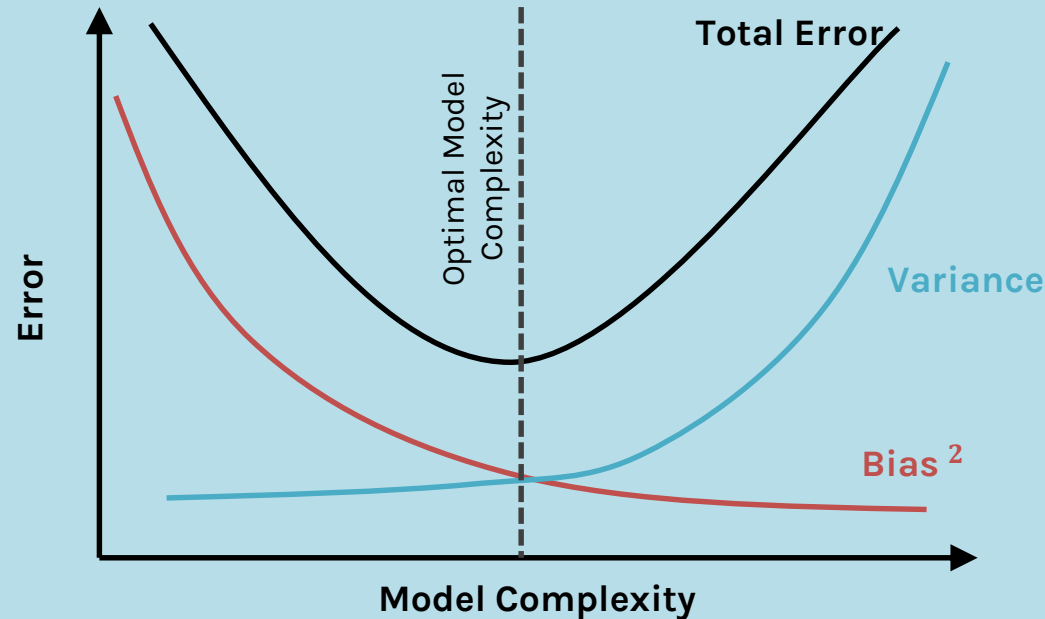
Please take a look at the following visualizations. Discuss with your neighbor: which one of the following models has the highest / lowest variance / bias.



How would a model with a good balance between variance and bias look like?

The bias-variance trade-off

- We need to make a trade-off between “too specific” and “too general”



1.3 Clustering

- Basic concept of clustering
- Measuring, metrics and similarity
- Clustering techniques

Most common business analytics jobs

Problem	Business Perspective	Techniques
Find Clusters/Outliers	<ul style="list-style-type: none"> Are there different types of users Can we put different products together into distinct/different groups? 	<ul style="list-style-type: none"> Clustering Outlier-Analysis
Find Relationships	<ul style="list-style-type: none"> If a customer buys product A, what does he buy next? Which product sets belong together? 	<ul style="list-style-type: none"> Association Analysis
Predict Classes	<ul style="list-style-type: none"> Is this customer solvent or not? Will this customer send back this shipping or not? 	<ul style="list-style-type: none"> Decision Trees Logistic Regression Support-Vector Machines
Predict Values	<ul style="list-style-type: none"> Does a new label increase the? Is there a relationship between Sales and Commercials? 	<ul style="list-style-type: none"> Regression Support-Vector Machines
Predict Developments	<ul style="list-style-type: none"> How will the value of our products develop? What future developments are likely? 	<ul style="list-style-type: none"> Time-Series Forecasting

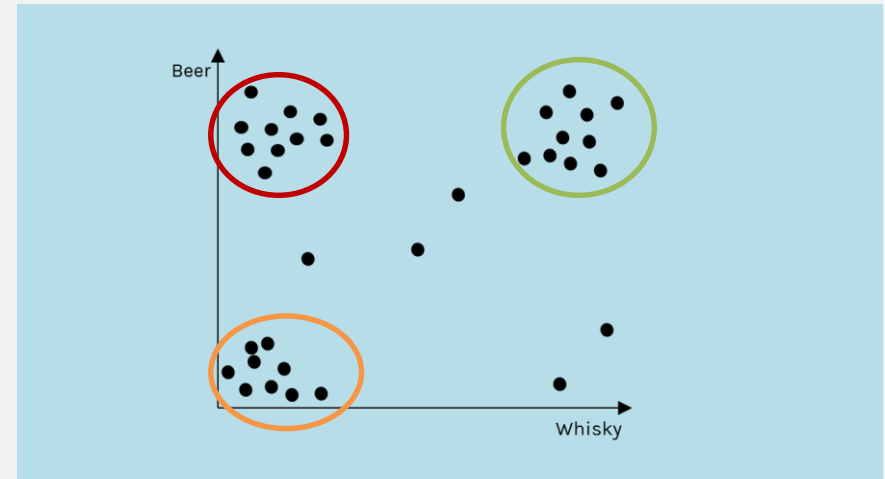


There will be further advanced techniques we will discuss in „Advanced Analytics with R“, another lecture covering topics like e.g. ANN, FP Growth techniques, Expectation Maximization etc.

Clustering

► **Forming groups of objects in a way that the same group (clusters) are more similar to each other than to those in other groups**

- Clustering is an unsupervised analytics technique
- Data structuring tool generally used for exploratory rather than confirmatory analysis
- Techniques differ significantly in their understanding of what constitutes a cluster



Use Cases:

- **Clustering users:** Finding users with similar interests and preferences for recommendations, e.g. Netflix or Amazon
- **Data Stream Clustering:** Cluster different telephone records, multimedia data, financial transactions, user log-files etc. for further analysis

What is clustering about?

- Grouping a set of data objects into clusters
- Cluster: a collection of data objects, similar to one another within the same cluster, and dissimilar to the objects in other clusters
- Clustering = unsupervised “classification” (no predefined classes)
- Typical usage in analytics:
 - As a stand-alone tool to get insight into data distribution
 - As a preprocessing step for other algorithms

Consumer clustering

Identify groups of items which are similar to each other

- **Example:** *Dominik's Liquor Bar*



But what is similarity?

What is similarity?



What is similarity?



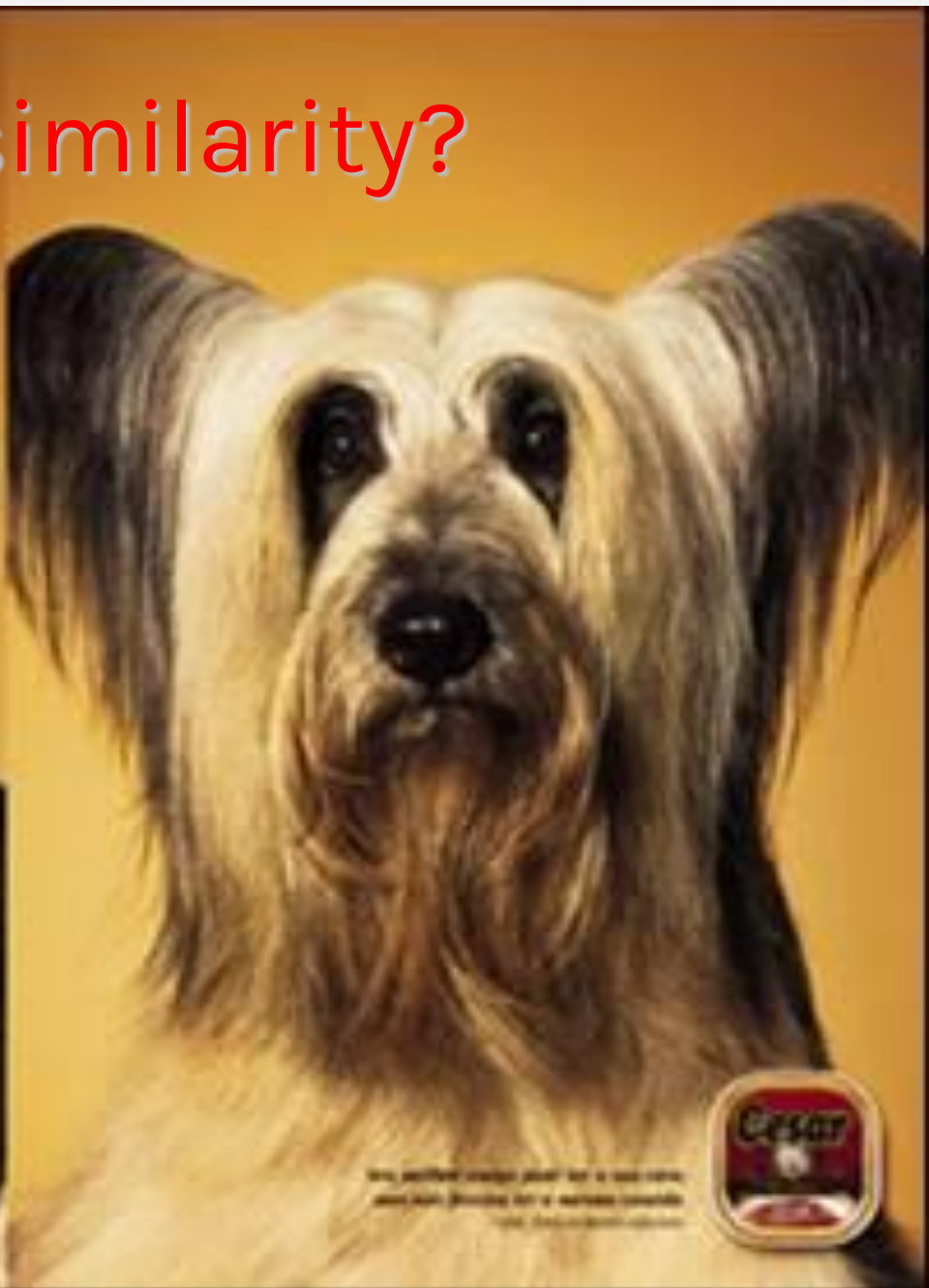
What is similarity?



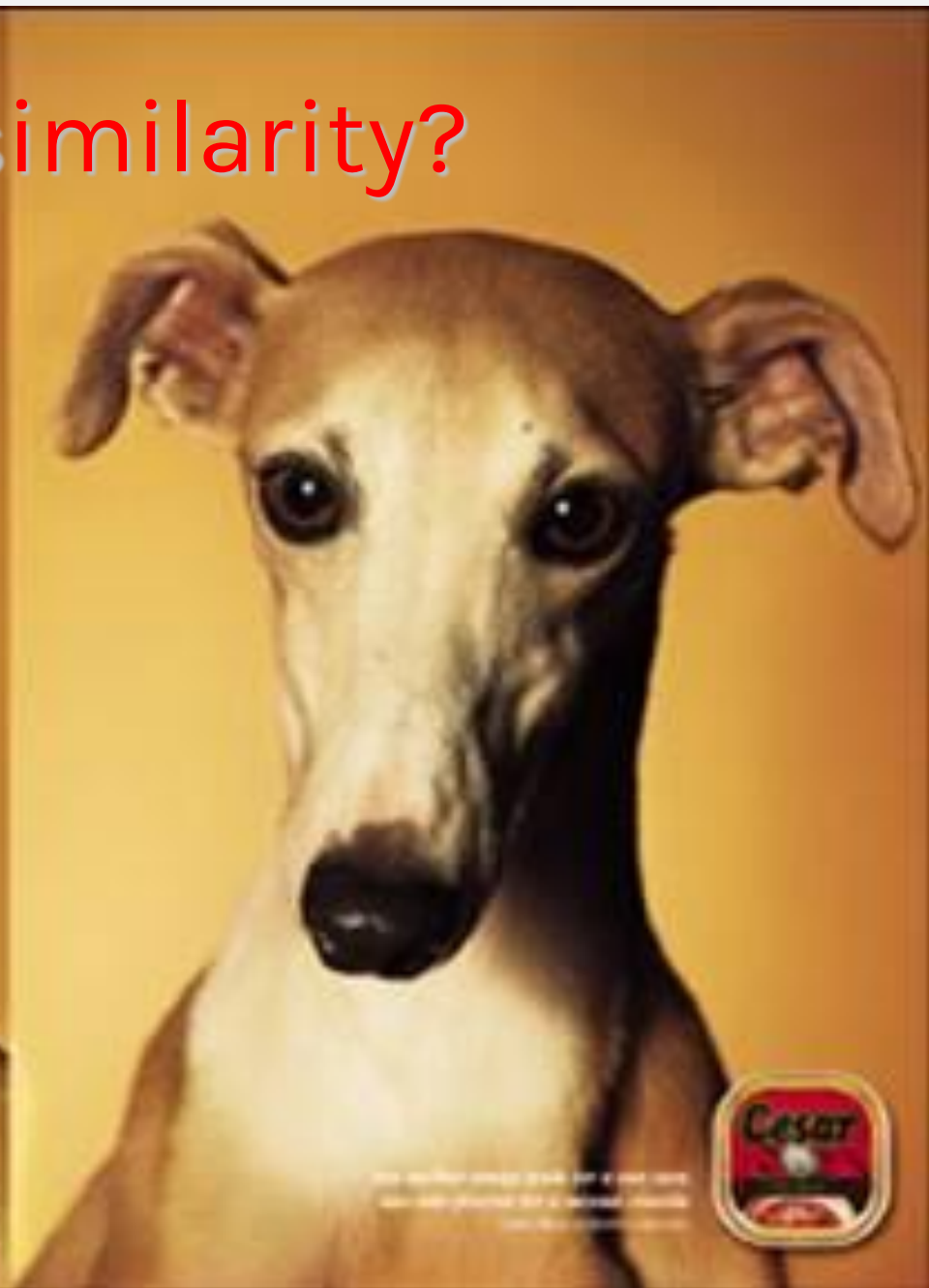
What is similarity?



What is similarity?



What is similarity?

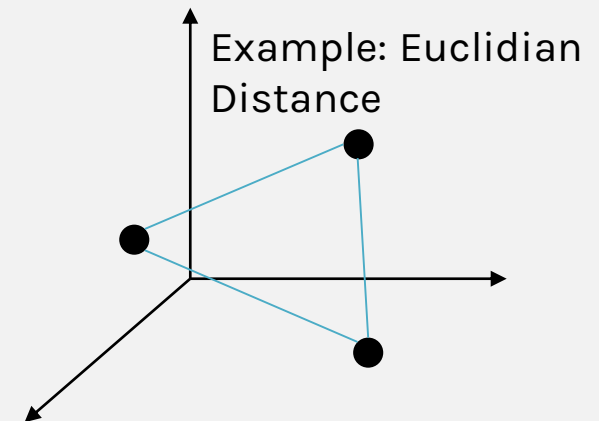


What is similarity?



Definition of a metric

- **Nonnegativity:** $\forall xy: d(x, y) \geq 0$
- **Symmetry:** $d(x, y) = d(y, x); s(x, y) = s(y, x)$
- **Triangle inequity:** $d(x, z) \leq d(x, y) + d(y, z)$
- **Reflexivity:** $d(x, y) = 0 \text{ iff. } x = y;$



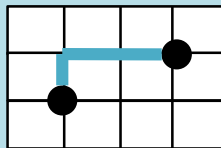
Minkowski Distance

$$d(x, y) = \left(\sum_{k=1}^n |x_k - y_k|^r \right)^{\frac{1}{r}}$$

- Generalization of Euclidian Distance

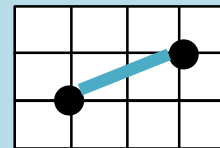
Mannhatten Distance

- $r = 1$
- Named Cityblock distance, or L_1 -norm
- For binary attributes also named Hamming Distance
- Sum of distance about all dimensions 1...n



Euclidian Distance

- $r = 2$
- L_2 -norm
- For numeric variables (ordinal or rational)
- Describes the geometric distance between two points



Supremum

- $r = \infty$
- L_∞ -norm
- Biggest difference in the dimensions





Difference L1 and L2 norm

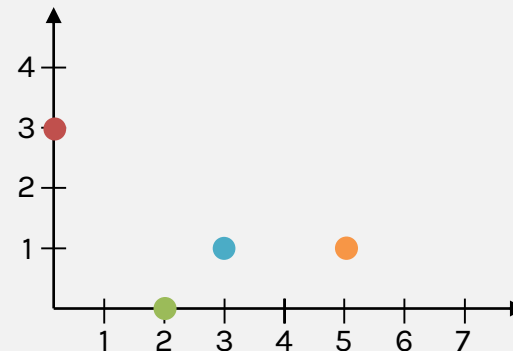
- L_1 : Impact of a difference proportional to difference itself
- L_2 : Higher relative impact of larger distances!
 - **Example:** $x = (3,3), y = (4,5)$
 - L1 norm distance: $|3 - 4| + |3 - 5| = 1 + 2 = 3$
 - L2 norm distance: $\sqrt{|3 - 4|^2 + |3 - 5|^2} = \sqrt{1 + 4} = 2.2$
- L_2 norm is smaller than L_1 but the individual difference has more relative weight

Your turn!





Task

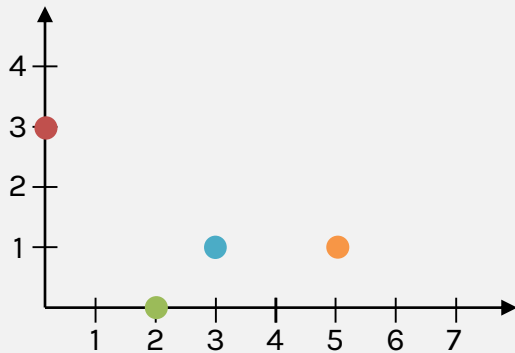
Please compute the Mankowski distances (L_1 , L_2 , and L_3) for the following dataset:

Observation	X	Y
	0	2
	2	0
	3	1
	5	1











Classroom task

Observation	X	Y
	0	2
	2	0
	3	1
	5	1



$$d(x, y) = \left(\sum_{k=1}^n |x_k - y_k|^r \right)^{\frac{1}{r}}$$







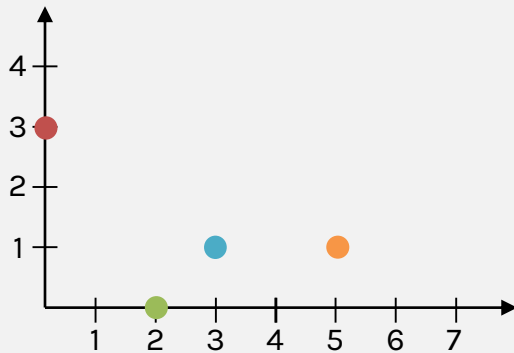
L_1				
				
				
				
				

$$d(x, y) = (|(0 - 0)|^1 + |(2 - 2)|^1)^{\frac{1}{1}}$$









$$d(x, y) = (|0 - 2| + |2 - 0|) = 2 + 2 = 4$$









Classroom task









Observation	X	Y
	0	2
	2	0
	3	1
	5	1



$$d(x, y) = \left(\sum_{k=1}^n |x_k - y_k|^r \right)^{\frac{1}{r}}$$

L_1				
	0	4	4	6
	4	0	2	4
	4	2	0	2
	6	4	2	0

L_2				
	0	2.828	3.162	5.099
	2.828	0	1.414	3.162
	3.162	1.414	0	2
	5.099	3.162	2	0

L_3				
	0	2	3	5
	2	0	1	3
	3	1	0	2
	5	3	2	0

Beyond Minkowski

Binary Objects

- Simple Matching Coefficient (SMC)
- Jaccard Coefficient

Correlation

- Pearson
- Speerman

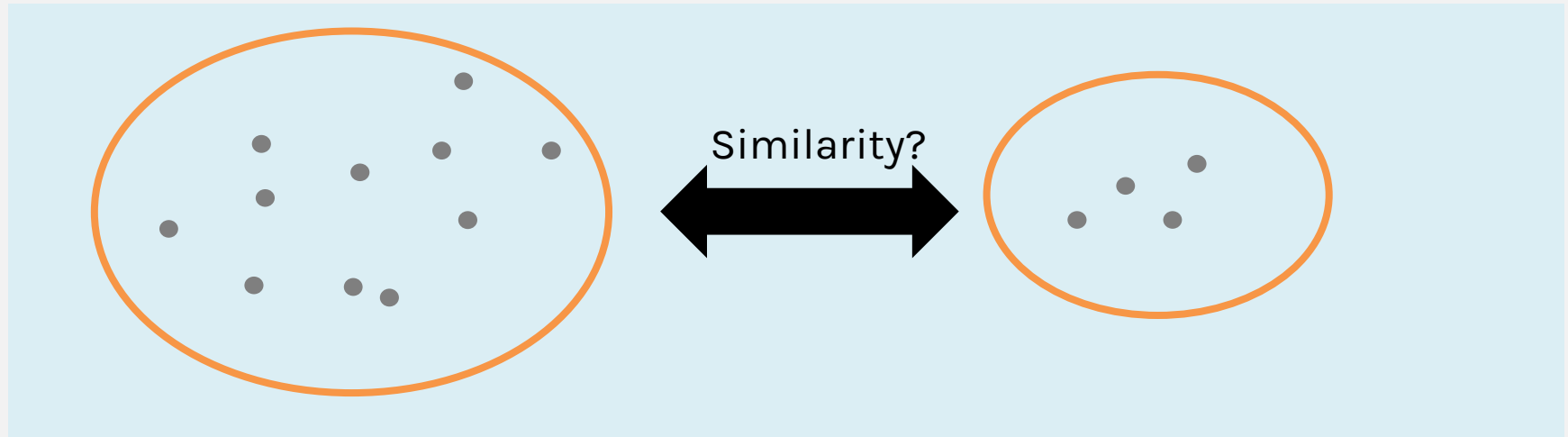
Vectors

- Simple Matching Coefficient
- Cosinus Coefficient
- Dice Coefficient
- Tanimoto Coefficient
- Overlap Coefficient



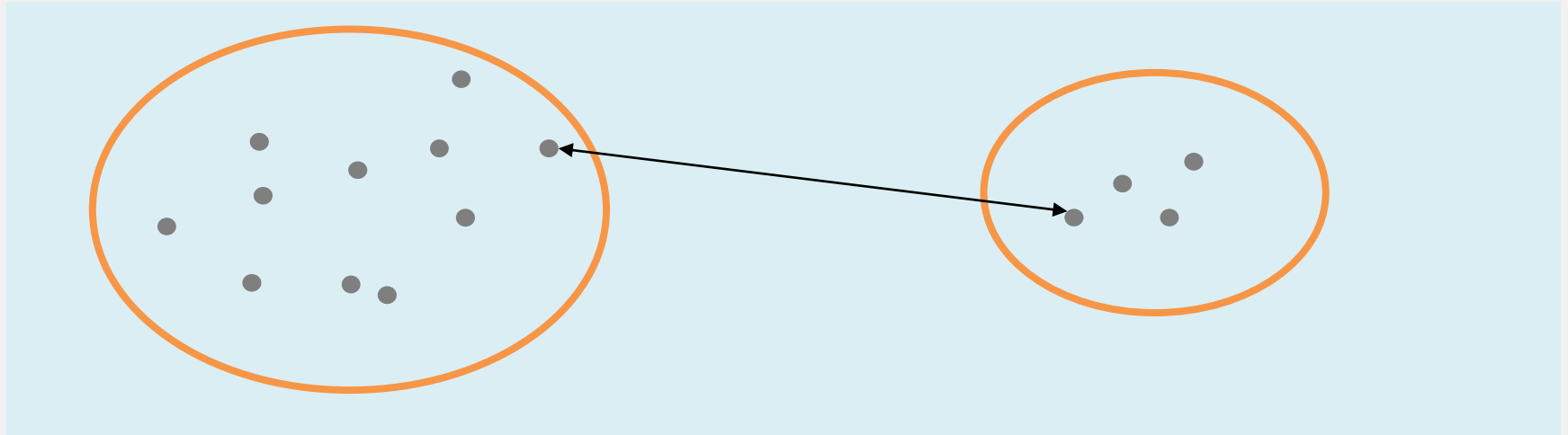
You find a comprehensive overview of further similarity concepts at „Cha, S. H. (2007). Comprehensive survey on distance/similarity measures between probability density functions”

Inter-Cluster Similarity



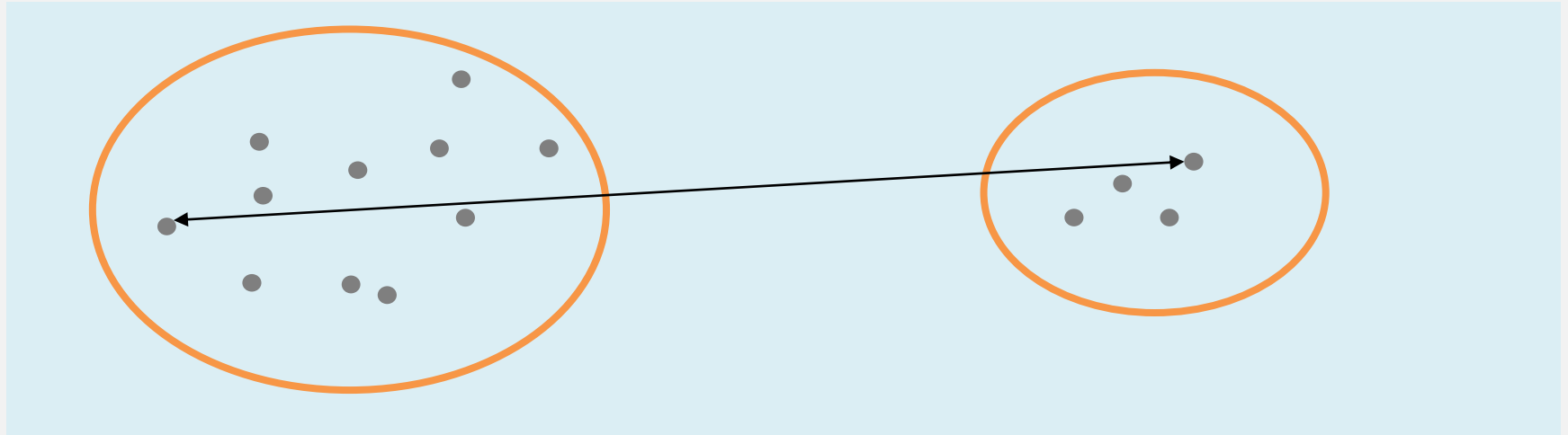
- MIN (Single Linkage)
- MAX (Group Linkage)
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

Inter-Cluster Similarity



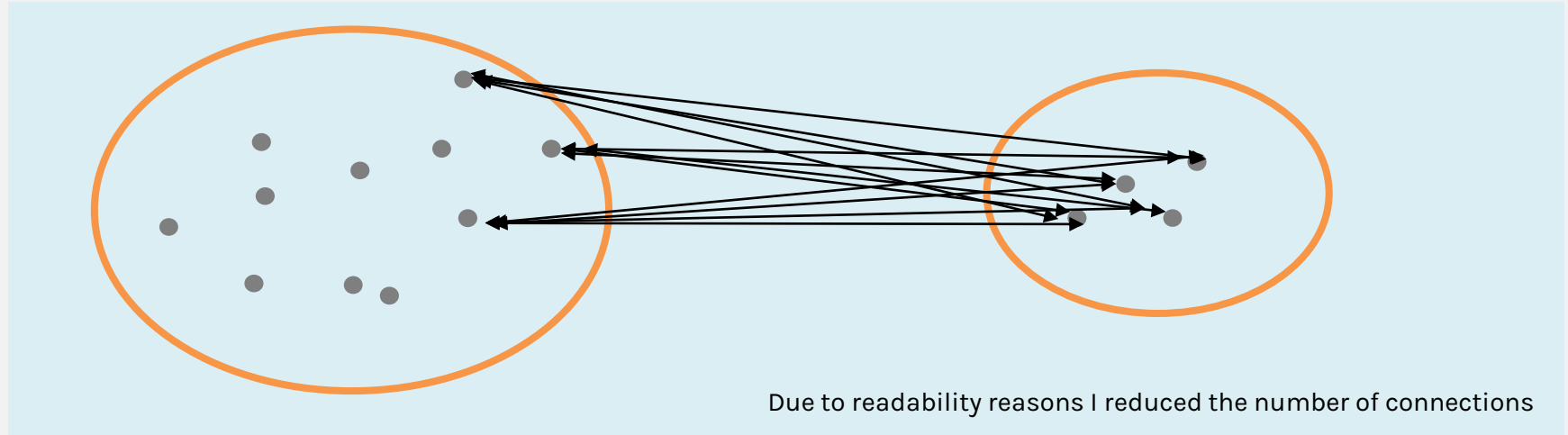
- MIN (Single Linkage)
- MAX (Group Linkage)
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

Inter-Cluster Similarity



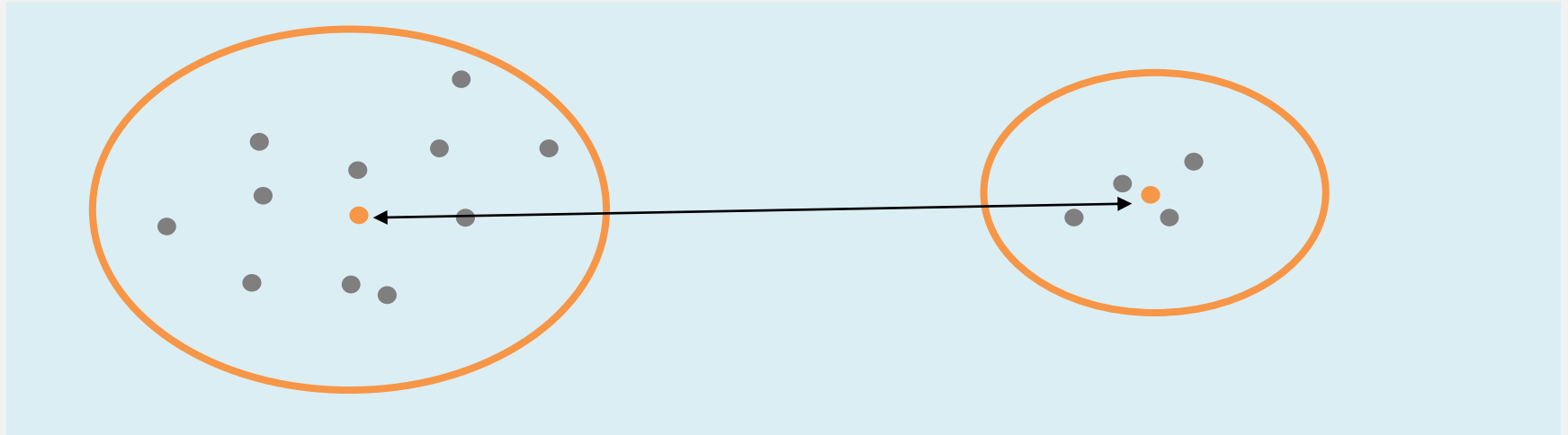
- MIN (Single Linkage)
- MAX (Group Linkage)
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

Inter-Cluster Similarity



- MIN (Single Linkage)
- MAX (Group Linkage)
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

Inter-Cluster Similarity



- MIN (Single Linkage)
- MAX (Group Linkage)
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
 - Ward's Method uses squared error

Cluster Similarity: Ward's Method

- **Similarity of two clusters is based on the increase in squared error when two clusters are merged**
 - Similar to group average if distance between points is distance squared
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of K-means
- Can be used to initialize K-means

A short taxonomy of clustering methods

- Partitioning algorithms
 - Find k partitions, minimizing some objective function
- Probabilistic Model-Based Clustering (EM)
- Density-based
 - Find clusters based on connectivity and density functions
- Hierarchical algorithms
 - Create a hierarchical decomposition of the set of objects
- Other methods
 - Grid-based
 - Neural networks (SOM's)
 - Graph-theoretical methods
 - Subspace Clustering

Partitioning Clustering: Basic Concept

- Approach: Group objects into a subset of clusters minimizing an objective function
- Problem: Exhaustively enumerating all possible partitions into k sets in order to find the global minimum is too expensive.

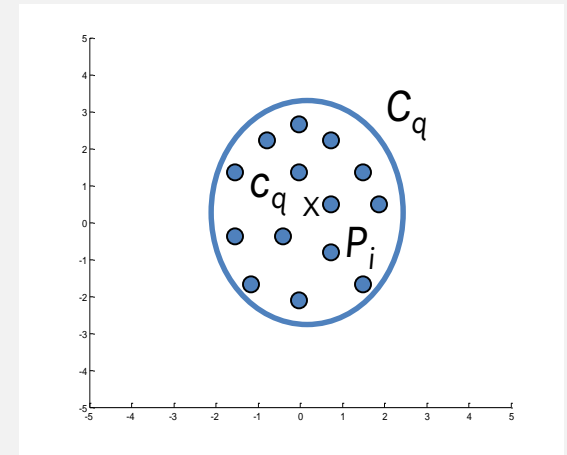


Heuristics

centroid and radius of a cluster

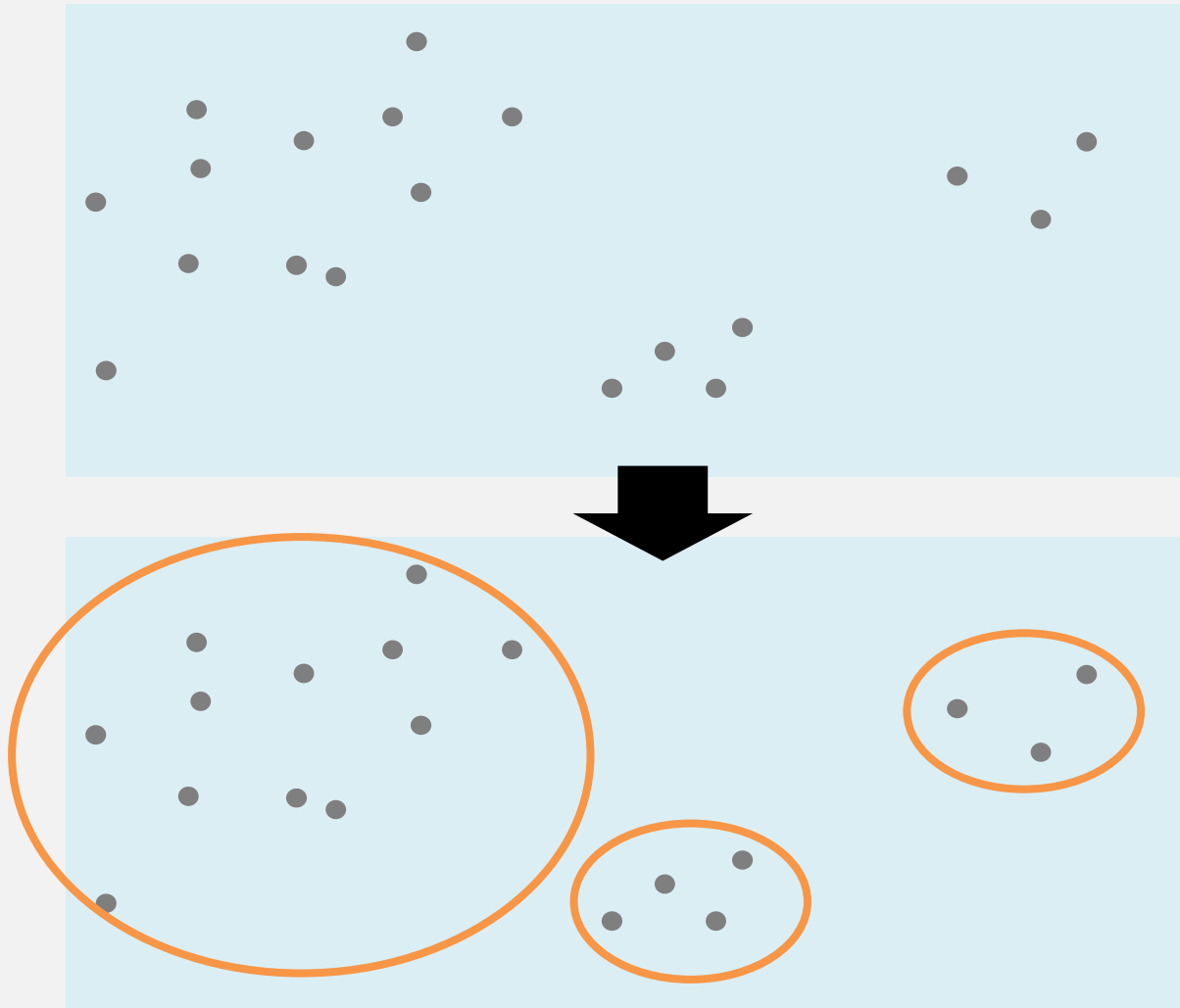
- Centroid c_q : “middle” of cluster C_q
- compute centroid c_q of C_q per dim. as:
- Cluster radius:** square root of average squared distance from any point of the cluster to centroid
- That is what we want to minimize (over all clusters)

$$c_q = \frac{\sum p_i}{|C_q|}$$



$$R_q = \sqrt{\frac{\sum (p_i - c_q)^2}{|C_q|}}$$

Partitioning



Initialization:

Choose k representatives for clusters, e.g., randomly

Repeat:

- Assign each object to the cluster it “fits best” in the current clustering
- Compute new cluster representatives based on these assignments
- Repeat until the change in the objective function from one iteration to the next drops below a threshold

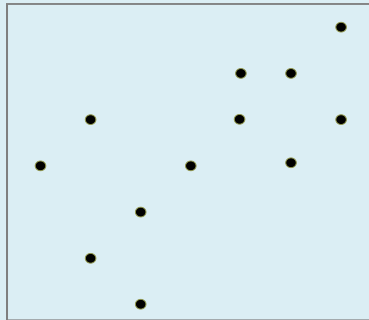
Partitioned Clustering (k-Means)

- Given a set of observations (x_1, \dots, x_n) , k -means clustering aims to partition the n observations into $k \leq n$ sets $S = \{S_1, \dots, S_k\}$ so as to minimize the within-cluster sum of squares and use the centroid of a cluster as representative
- Or more formally:

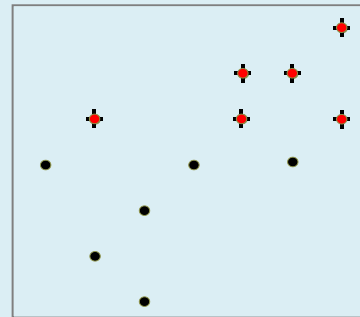
$$\arg \min_S \sum_{j=1}^k \sum_{x_i \in S_j} \|x_i - \bar{m}_j\|^2.$$

Example k-means clustering with $k = 2$

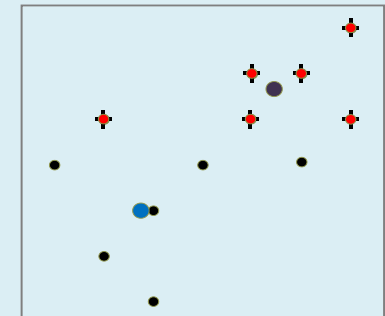
Initial data set



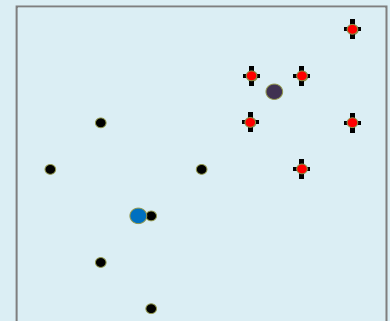
1. Arbitrarily partition objects into k groups



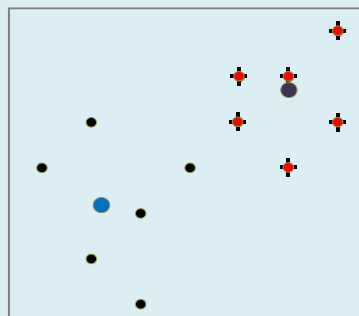
2. Update cluster centroids



3. Reassign objects



4. Update cluster centroids



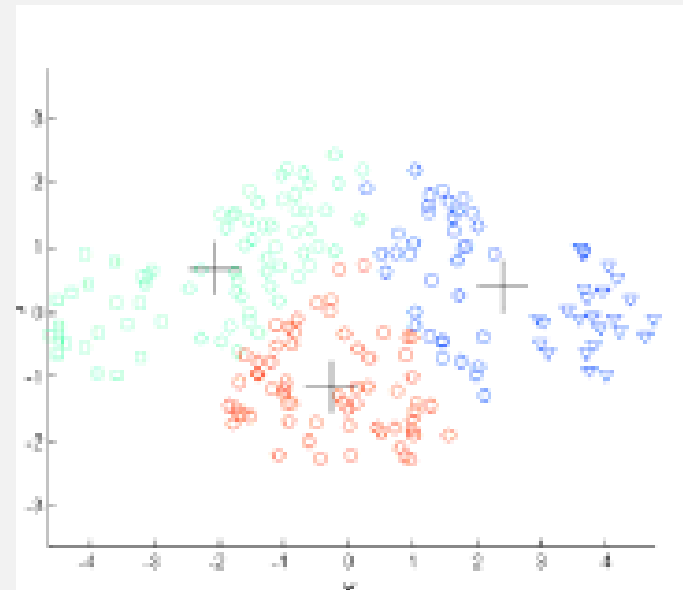
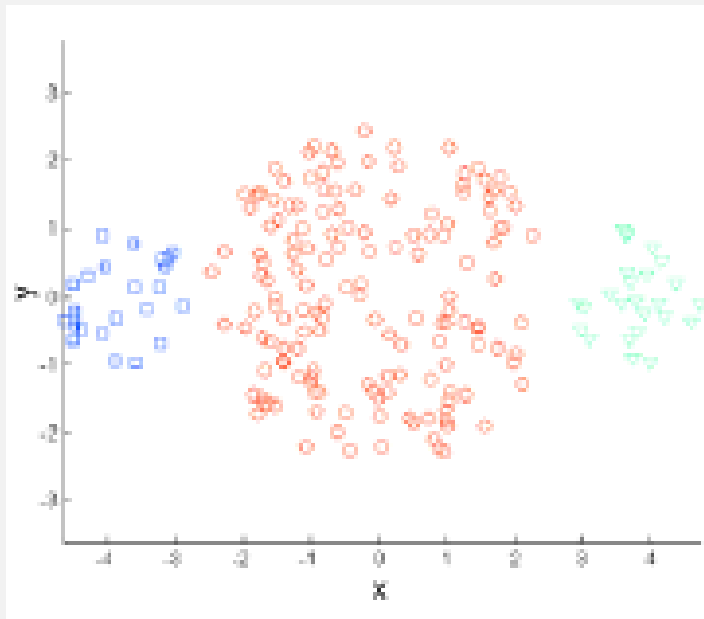
Iterate if assignment changed

- **Terminate if assignment does not change** or if no significant reduction in **total squared distances**

- $\rightarrow C_1, \dots, C_k$

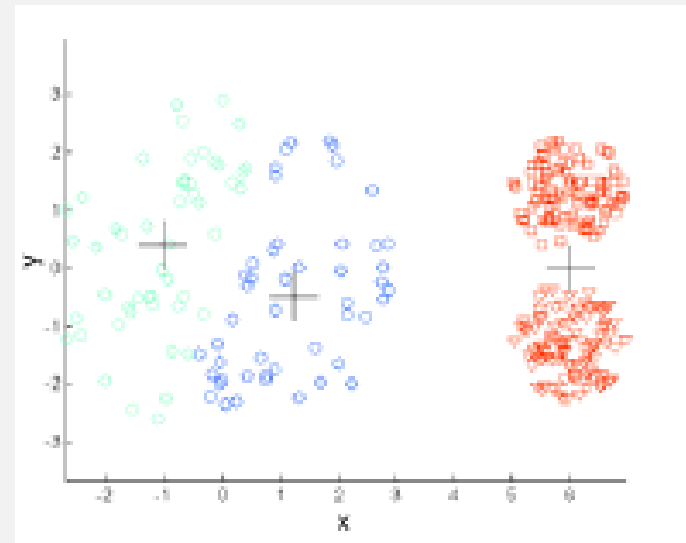
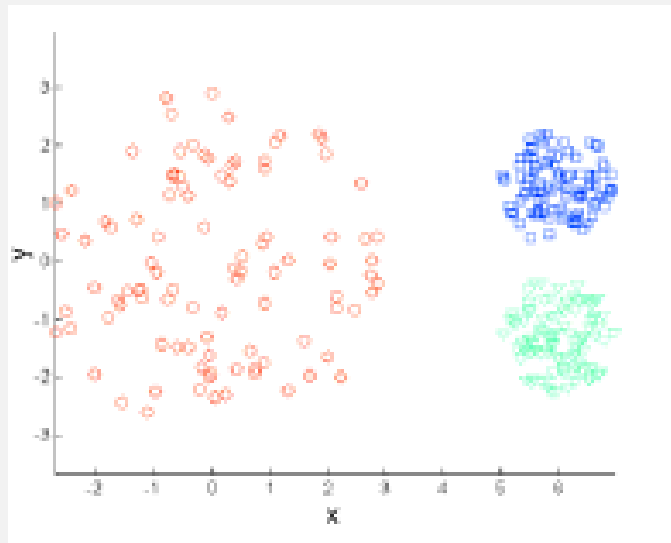
Problems with kmeans

- Large natural clusters tend to be scattered by kmeans



Problems with kmeans

- Natural clusters are also not always found due to their density



- Problems with K-Means:
 - Applicable only when mean is defined
 - Outliers have a strong influence on the result
- The influence of outliers is intensified by the use of the squared error use the absolute error (total distance instead): $TD(C) = \sum_{p \in C} dist(p, m_c)$ and $TD(C) = \sum_{C \in \mathcal{C}} TD(C_i)$
- Three alternatives:
 - Medoid: representative object “in the middle”
 - Mode: value that appears most often
 - Median: (artificial) representative object “in the middle”

Clustering with kmeans in

`kmeans()`

```
kmeans(x, centers=x, iter.max=100, algorithm="Lloyd")
```

Parameters

<code>x</code> (Input)	Numeric matrix or data frame that can be coerced to such a matrix (data frame with all numeric columns)
<code>centers</code>	Either the number of clusters, say <code>k</code> , or a set of initial distinct cluster centers
<code>iter.max</code>	The maximum number of iterations allowed
<code>algorithm</code>	Options available are: Hartigan-Wong, Lloyd, Forgy, or MaxQueen

kmeans output

```
> fit
```

```
K-means clustering with 3 clusters of sizes 7, 62, 17
```

```
Cluster means:
```

	Body	Sweetness	Smoky	Medicinal	Tobacco
1	1.4604351	-1.2019264	2.1927500	2.6224854	1.8546327
2	0.1330391	0.3592557	-0.1150977	-0.2424769	-0.1605354
3	-1.0865570	-0.8153156	-0.4831291	-0.1955193	-0.1781902

the coordinates of the cluster centers

```
Clustering vector:
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
2	2	2	1	2	2	3	2	2	2	2	2	2	2	2	2	3	2	2	3	3	1	2	1	2	2	2	2	2	2	2	2	2	
35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68
3	3	2	3	2	2	2	2	2	2	2	2	3	3	2	3	2	2	2	2	3	2	1	1	2	3	2	2	3	2	2	2	3	
69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86																
1	2	2	2	2	2	2	2	2	1	3	2	2	3	2	3	2	2																

the cluster of each value of the input data

```
Within cluster sum of squares by cluster:
```

```
[1] 31.29875 164.44759 46.64205  
(between_SS / total_SS = 43.0 %)
```

sum of squared errors

```
Available components:
```

[1]	"cluster"	"centers"	"totss"	"withinss"	"tot.withinss"	"betweenss"
[7]	"size"	"iter"	"ifault"			

- k-means is fast, simple, and the most widely used clustering algorithm
- However, k-means not guaranteed to find the optimal solution
- Invoking algorithm using variety of initial cluster centers improves probability of achieving best result

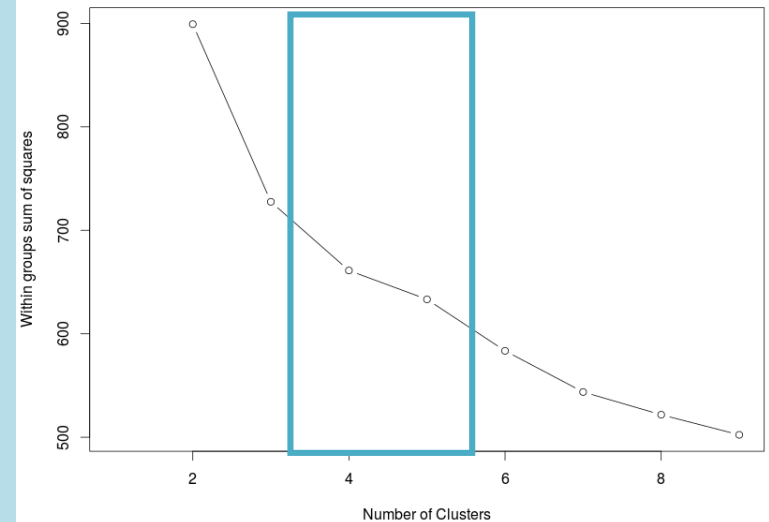
But: What is an appropriate value for k?

- Potential problem for applying k-means
- Analyst may have *a priori* knowledge of k
- Or: Cycling through different k values, selecting “best” solution
- Or: *Hierarchical Clustering might help here!*

Elbow criterion

- Choose a number of clusters so that adding another cluster doesn't give much better modeling of the data

```
ssplot = function(data, maxCluster = 9) {  
  SSw = vector()  
  for (i in 2:maxCluster) {  
    SSw[i] = sum(kmeans(data, centers =  
i)$withinss)  
  }  
  plot(1:maxCluster, SSw, type = "b",  
xlab = "Number of Clusters", ylab =  
"Within groups sum of squares")  
}  
  
# Plot the Sum of Squares  
ssplot(whiskies_k)
```



- Run k-means clustering on the dataset for a range of values of k (say, k from 1 to 10 in the examples above)
- For each value of k calculate the sum of squared errors (SSE)
- Plot a line chart of the SSE for each value of k
- If the line chart looks like an arm, then the "elbow" on the arm is the value of k that is the best

A short taxonomy of clustering methods

- Partitioning algorithms
 - Find k partitions, minimizing some objective function
- Probabilistic Model-Based Clustering (EM)
- Density-based
 - Find clusters based on connectivity and density functions
- Hierarchical algorithms
 - Create a hierarchical decomposition of the set of objects
- Other methods
 - Grid-based
 - Neural networks (SOM's)
 - Graph-theoretical methods
 - Subspace Clustering

Hierarchical Clustering

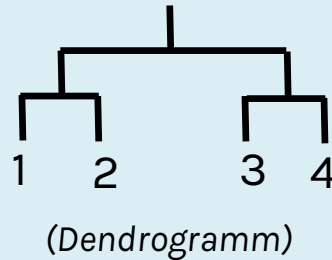
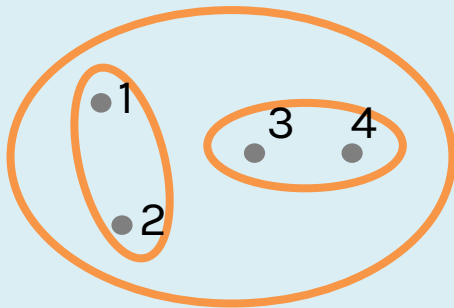
- Do not have to assume any particular number of clusters
- Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level
- They may correspond to meaningful taxonomies
- Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, ...)

Hierarchical Clustering

- Method of cluster analysis which seeks to build a hierarchy of clusters
 - **Agglomerative:** "bottom up" approach; each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy
 - **Divisive:** "top down" approach; all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy
- Traditional hierarchical algorithms use a similarity or distance matrix
- Merge or split one cluster at a time

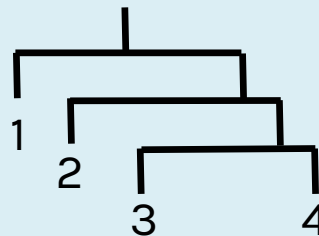
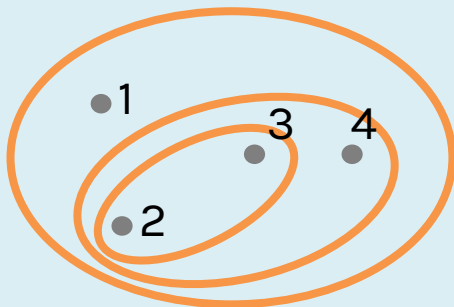
Hierarchical Clustering

Traditional hierarchical



- Produces a set of nested clusters organized as hierarchical tree
- Can be visualized as a dendogram

Hierarchical constrained



- Basic algorithm is straightforward
 - 1. Compute the proximity matrix
 - 2. Let each data point be a cluster
 - **3. Repeat**
 - 4. Merge the two closest clusters
 - 5. Update the proximity matrix
 - **6. Until only a single cluster remains**
- **Key operation is the computation of the proximity of two clusters**
- Different approaches to defining the distance between clusters distinguish the different algorithms

1.4 Association Analysis

- Frequent item sets
- Association rules
- Association rule learning

Most common business analytics jobs

Problem	Business Perspective	Techniques
Find Clusters/Outliers	<ul style="list-style-type: none"> Are there different types of users Can we put different products together into distinct/different groups? 	<ul style="list-style-type: none"> Clustering Outlier-Analysis
Find Relationships	<ul style="list-style-type: none"> If a customer buys product A, what does he buy next? Which product sets belong together? 	<ul style="list-style-type: none"> Association Analysis
Predict Classes	<ul style="list-style-type: none"> Is this customer solvent or not? Will this customer send back this shipping or not? 	<ul style="list-style-type: none"> Decision Trees Logistic Regression Support-Vector Machines
Predict Values	<ul style="list-style-type: none"> Does a new label increase the? Is there a relationship between Sales and Commercial? 	<ul style="list-style-type: none"> Regression Support-Vector Machines
Predict Developments	<ul style="list-style-type: none"> How will the value of our products develop? What future developments are likely? 	<ul style="list-style-type: none"> Time-Series Forecasting



There will be further advanced techniques we will discuss in „Advanced Analytics with R“, another lecture covering topics like e.g. ANN, FP Growth techniques, Expectation Maximization etc.

Association rule learning

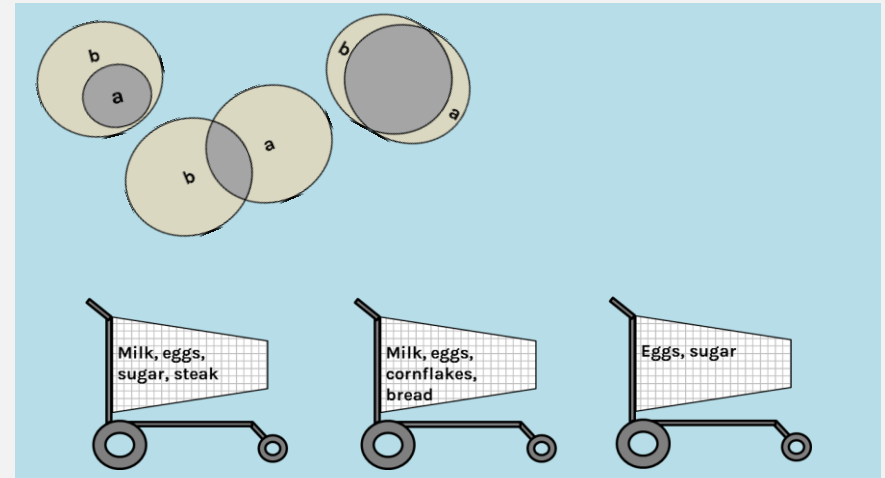
➤ Describes how specific phenomena relate to each other.

- Used to discover interesting regularities between variables
- The uncovered relationships can be represented in form of association rules
- does not consider the order of items either within a transaction or across transactions.

Example:

$\{\text{Whisky, Potatoes}\} \rightarrow \{\text{Steak}\}$

Customers buying whisky and potatoes together are also likely to buy a steak.



Use Cases:

- **Shopping basket analysis:** Identify customer preferences by finding associations and correlations between different products that customers bought together.
- **(Web) Pattern Mining:** use multi-dimensional association rules to identify behavioral patterns based on log files.

Frequent item sets: data structure

- Item: one single item
- Item set: a set of single items
- k-item set: An item set that contains k items
- Transaction:

Relational structure

<Tid, item>
<1, item A>
<1, item B>
<2, item C>

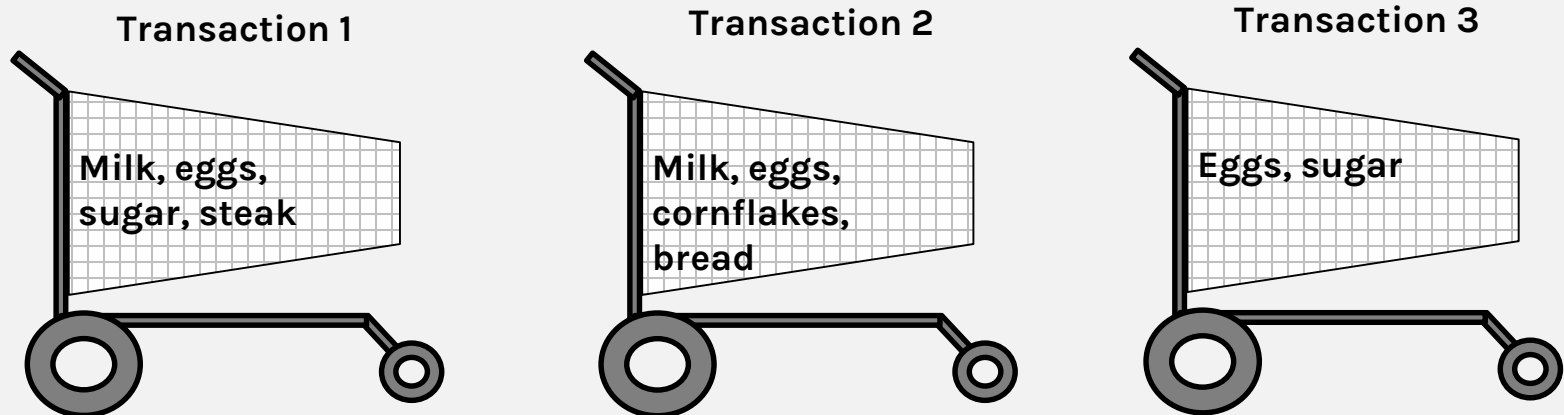
Compact structure

<Tid, item>
<1, {item A, item B}>
<2, {item C}>

- In shopping basket analysis, the items are the goods; a transaction is a single purchase; the term *transaction* has nothing to do with the classic database concept.

Frequent item sets

- **Example:** Basket analysis



- **Support of an item set I:**
 - Number of transactions containing item
- **Minimum support σ :**
 - Support Threshold



Frequent item set

An item set with support $\geq \sigma$

Example: Frequent item set

Transaction ID	Items
1	Whisky, lemon
2	Whisky, lemon, cucumber
3	Whisky
4	Lemon, chips

- **Support:**

- $\text{Support}(\text{whisky}) = 3$ (75%)
- $\text{Support}(\text{lemon}) = 3$ (75%)
- $\text{Support}(\text{whisky, lemon}) = 2$ (50%)

- *If $\sigma = 60\%$, then:*

- $\{\text{whisky}\}$ and $\{\text{lemon}\}$ are frequent, but $\{\text{whisky, lemon}\}$ is not



Support (item set)

Fraction of transactions that contain an itemset

Consideration: Support and item sets

- A specific frequent item set is maximal, iff. it is not a subset of another frequency item set
- It is only necessary to explicitly generate the maximal frequency item sets in order to know the frequency item sets.
- Does not apply to Association Rules (we will come to that later)

Expansion of frequent itemsets

- **Association rule:** an implication expression of the form $X \rightarrow Y$, where X and Y are item sets.
- They are represented in the form of two-sided rules

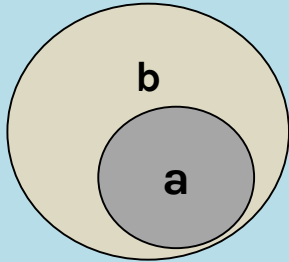
$$LHS \rightarrow RHS [support, confidence]$$

- the left-hand side (LHS) implies the right-hand side (RHS), with a given value of support and confidence
- support and confidence are measures of the quality of a given rule

Frequent itemsets and association rules

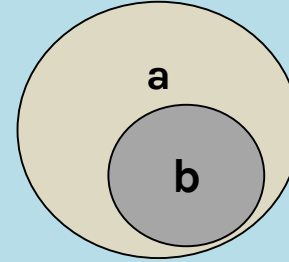
1

$a \rightarrow b$



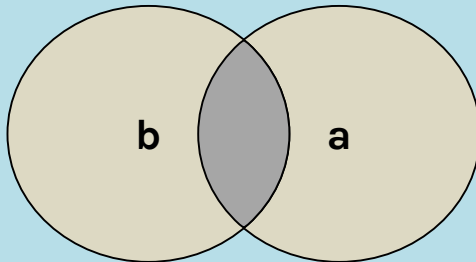
2

$b \rightarrow a$



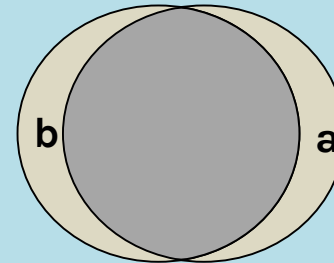
3

Very low relationship



4

Almost $a \rightarrow b, b \rightarrow a$



Frequent item sets like $I = \{a, b\}$ make no difference between 1,2, and 4 - but rules do!

Selection criteria for association rules

- Given two item sets A and B ,
- and the association (rule) $A \rightarrow B [s, c]$

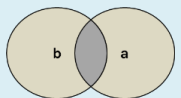
D s : support of $A \rightarrow B = \text{support}(A \cup B)$

- Number of sets containing A and B .
- Support was already defined for sets, is now defined for rules

D c : confidence of $A \rightarrow B = \frac{\text{support}(A \cup B)}{\text{support}(A)}$

?

If you take a look at the set illustrations from the previous slide, where is the $\text{support}(A \cup B)$ in the illustrations, how would you draw it?



Selection criteria for association rules

- Selection criteria:
 - a) Minimum support: σ
 - b) Minimum confidence: γ
- We are interested in rules with $S > \sigma$, and $c > \gamma$

Support and confidence

- **Support: Frequency of the rule in the number of transactions.**
- $support(A \rightarrow B [s, c]) = support(A \cup B)$
- **Confidence: Proportion of transactions with A, which also contain B**
- $confidence(A \rightarrow B [s, c]) = p(B|A) = \frac{p(A \cap B)}{p(A)}$
- Support indicates how often a rule exists in the database
- Confidence indicates if the rules is true or not



Confidence would not be needed, if we only focus on frequency item sets

Example: support and confidence

Transaction ID	Items
1	Whisky, lemon
2	Whisky, lemon, cucumber
3	Whisky
4	Lemon, chips

- Frequent item set: Fraction of transactions that contain an itemset.
 - $Support(whisky, lemon) = \frac{2}{5}$
- Association rule: Fraction of transactions that contain both X and Y.
 - $Support(whisky \rightarrow lemon) = \frac{2}{5}$
 - $Confidence(whisky \rightarrow lemon) = \frac{\frac{2}{5}}{\frac{3}{5}} = \frac{2}{3}$

Your turn!

Task

Please calculate the support and confidence for the following rules

- *chips* → *cheese*
- *cheese* → *whisky*
- *sausages* → *ice cream*

$$\text{support of } A \rightarrow B = (A \cup B)$$
$$\text{confidence of } A \rightarrow B = \frac{\text{support } (A \cup B)}{\text{support } (A)}$$

ID	Items
T1	{sausages, ice cream, whisky}
T2	{ice cream, whisky}
T3	{chips, cheese}
T4	{sausages, whisky, cheese}
T5	{whisky, chips, cheese}

Classroom task

Rule	Confidence	Support
<i>chips</i> → <i>cheese</i>	100 %	40 %
<i>cheese</i> → <i>whisky</i>	66.6 %	40 %
<i>sausages</i> → <i>ice cream</i>	50%	20 %



Why there is a difference between
chips → *cheese*, and *cheese* → *chips*?

ID	Items
T1	{sausages, ice cream, whisky}
T2	{ice cream, whisky}
T3	{chips, cheese}
T4	{sausages, whisky, cheese}
T5	{whisky, chips, cheese}

Parameter specifications

- Minimum support σ
 - **High:** few frequent item sets, few frequent rules
 - **Low:** many true rules, but they are not frequent

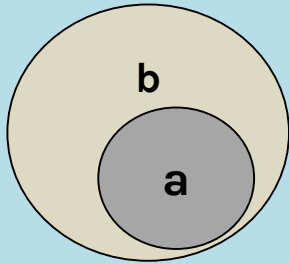
- Minimum confidence: γ
 - **High:** few rules, but mostly true
 - **Low:** many rules, but they are probably not true

- Best-Practice:
 - a) $\sigma = 2 - 10 \%$
 - b) $\gamma = 70 - 90 \%$

Visualization of association rules

1

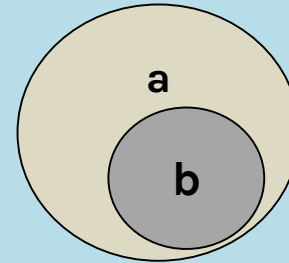
$\text{confidence}(a \rightarrow b) \approx 100\%$
 $\text{confidence}(b \rightarrow a) \approx 0\%$



Assumption: a is much smaller than B

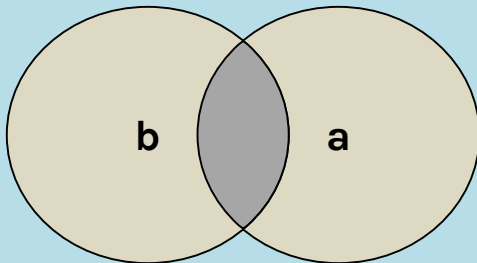
2

$\text{confidence}(a \rightarrow b) \approx 0\%$
 $\text{confidence}(b \rightarrow a) \approx 100\%$



3

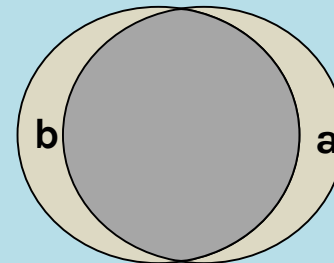
$\text{confidence}(a \rightarrow b) \approx 0\%$
 $\text{confidence}(b \rightarrow a) \approx 0\%$



Assumption: the inter-section is very small

4

$\text{confidence}(a \rightarrow b) \approx 100\%$
 $\text{confidence}(b \rightarrow a) \approx 100\%$



Recapitulation

- If everyone buys whisky and beer together, then the association rule “beer \rightarrow whisky” has...
 - a) High support
 - b) High confidence
- If the customers stop to buy whisky, then the confidence is reduced

Misleading association rules

- However, an association rule that satisfies the minimum support s and the minimum confidence c does not necessarily need to be interesting!
 - **Example:** Customer buy beer and then whisky with $\text{support}(\text{beer} \rightarrow \text{whisky}) = 0.45$, buy whisky with $\text{support}(\text{whisky}) = 0.75$, and beer with $\text{support}(\text{beer}) = 0.6$. Then, confidence $(\text{beer} \rightarrow \text{whisky}) = \frac{0.45}{0.6} = 0.75$.
- The confidence in the example is exactly as large as the support of whisky. It does not make any statement about the association between beer and whisky and reflects only the $\text{support}(\text{whisky})$.

A further measure of interestingness: lift

- One way to address this problem is by applying a measure known as lift

$$\text{lift}(A \rightarrow B) = \frac{\text{confidence}(A \rightarrow B)}{\text{support}(B)} = \frac{\text{support}(A \cup B)}{\text{support}(A) \cdot \text{support}(B)}$$

- The lift compares the frequency of a pattern against a baseline frequency computed under the statistical independence assumption.
- The lift describes the correlation between binary items (> 1 positive correlation, <1 negative correlation, 1 no correlation)
- The lift is a symmetric measure: $\text{lift}(A \rightarrow B) = \text{lift}(B \rightarrow A)$

- Given the rule $\{Cola, Chips\} \rightarrow \{Crushed\ Ice\}$ has a lift of 1.25
- Then compared to item sets that are statistically independent of the crushed ice, we expect crushed ice to appear 25 % more often in item sets that contain cola and chips

Problems with lift

- The lift is sensitive with respect to the support of the items.
- Rare sets of items could generate high values of lift.
 - **Example:** suppose the rule $\{vegan\ chips\} \rightarrow \{Junglivet\ Whisky\}$ has a $support(vegan\ chips \cup Junglivet\ Whisky) = 0.01$, then $lift(vegan\ chips \rightarrow ice\ cream) = 4$
- One may think that this is an interesting rule, but only a very small number of customers buy vegan pizza.
- A marketing plan to encourage vegan pizza buyers to purchase Junglivet might not have a high impact.

Association rules in Business Analytics

- Extraction of information from consumer behavior
- Derive recommendations for e.g.
 - different layout of shops
 - other arrangement of the products
 - Change in product assortments
 - Special offers
- Can be used to find behavioral patterns in other areas
 - a) Credit card and transaction mining
 - b) Services from telecommunication provider
 - c) Finance and FinTech
 - d) Medical treatments
 - e) Web Analytics

Finding association rules

- **AIS algorithm (Agrawal, Imielinski and Swami, 1993)**
 - The first algorithm for generating simple association rules.
 - The potentially frequent sets are generated and counted "on the fly" by scanning the database (very computationally intensive).
- **SETM algorithm (Houtsma and Swami, 1993)**
 - Motivation: Use SQL to search for association rules.
 - The potential sets (analogous to the AIS algorithm) are generated based on transformations of the database.
 - The list of candidates is a separate step.
- **Disadvantages of AIS and SETM:**
 - The superfluous generation and enumeration of candidates, who later turn out to be not frequent
 - for the rest of my life.

Apriori Algorithm

Rakesh Agrawal and Ramakrishnan Srikant (1994)

- The Apriori algorithm for finding strong association rules is an iterative algorithm that is based on the Apriori principle

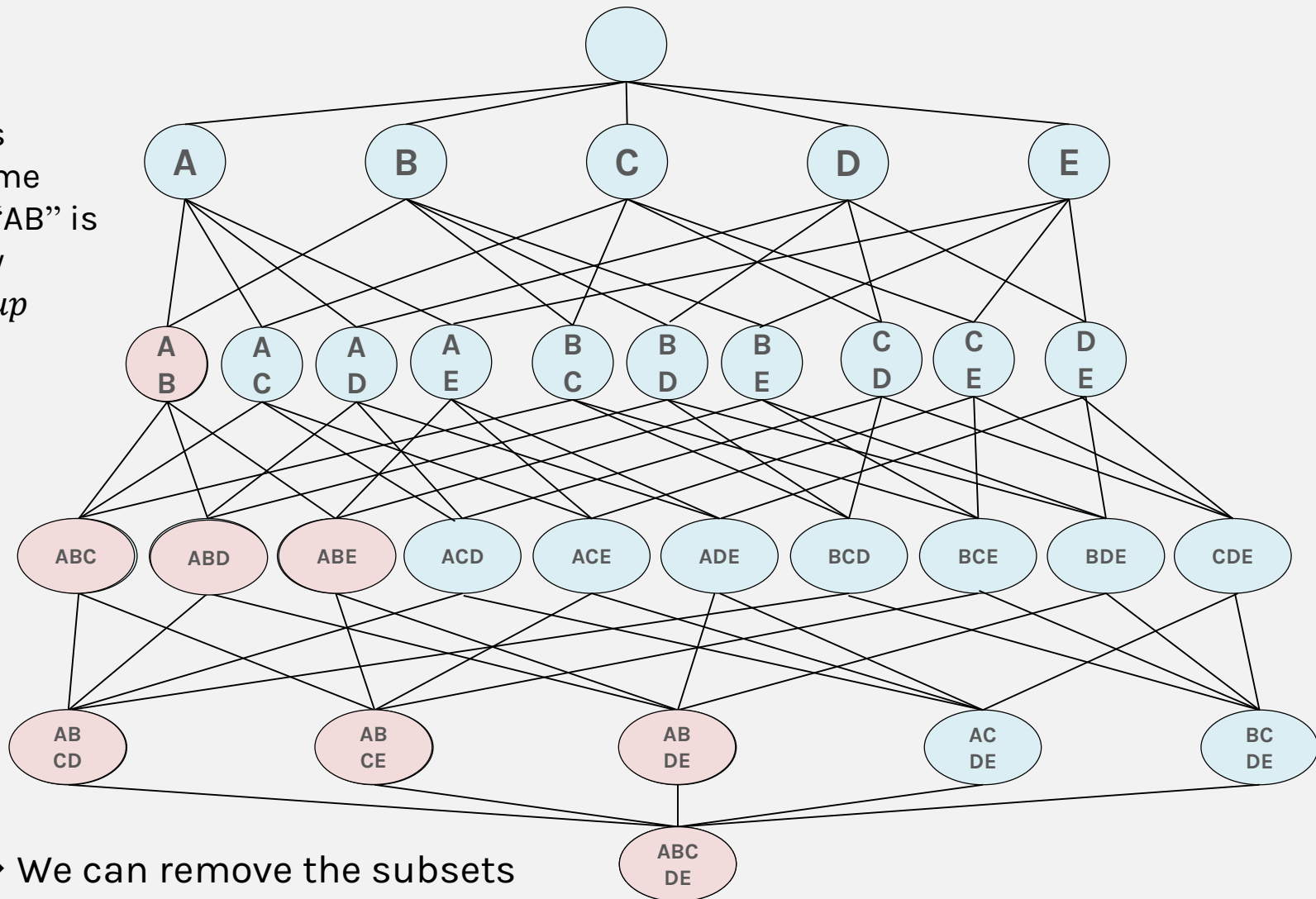
D Apriori principle

If an item set is frequent, then all of its subsets must also be frequent. Conversely, if an item set is infrequent, then all supersets are infrequent.

- **Frist step:** Searching for common patterns (frequent item sets) with the actual a priori Algorithm.
- **Second step:** Derive rules from the frequent sets.

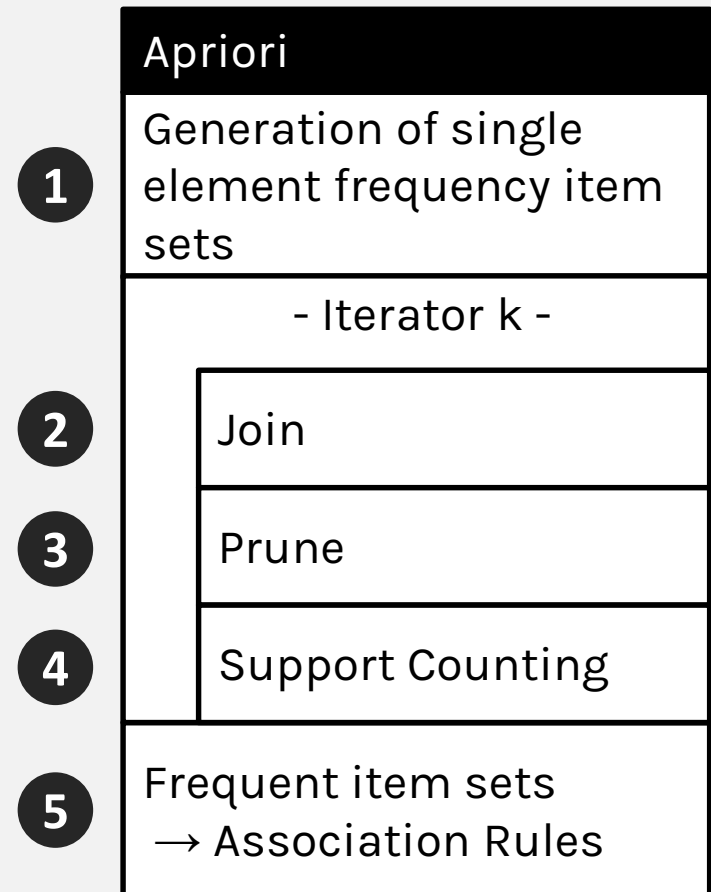
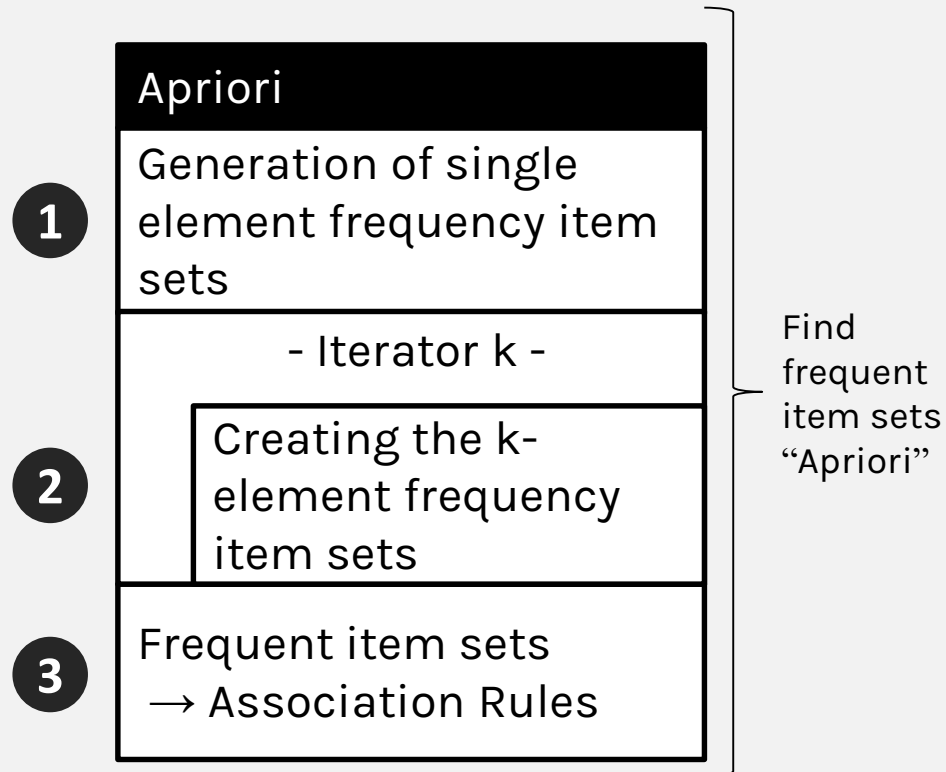
Apriori principle

Let us assume that “AB” is below *minsup*



→ We can remove the subsets

Apriori



Apriori in detail

- Calculation of the k sets from the $(k-1)$ sets
- **Join Step:** Determination of Candidates; A priori trick: All $(k-1)$ -elementary subsets of a k set are $(k-1)$ sets
- **Prune Step:** Delete all candidates, that contains an "inadmissible" $(k-1)$ elemental subset.
- **Support counting:** count how often the candidates really are

Pruning Example

Given $minsup = 0.6$, $\sigma_{min} = 3$

TID	Itemset
1	Brot Milch
2	Brot Windeln Eier Bier
3	Milch Windeln Bier Cola
4	Brot Milch Windeln Bier
5	Brot Milch Windeln Cola

Itemset	σ
Bier	3
Brot	4
Cola	2
Eier	1
Milch	4
Windeln	4



2 elm. Itemset	σ
Bier Brot	2
Brot Milch	2
Bier Windeln	3
Brot Milch	3
Brot Windeln	3
Windeln Milch	3



2 elm. Itemset	σ
Brot Milch Windeln	2

Result:

- 6 + 6 + 1 possible candidates
- Without pruning there would be 6 + 15 + 20 = 41

Apriori Algorithm

```
L1 = {large 1 – item sets};  
for (k = 2 ; Lk-1 ≠ {} ; k ++ ) do begin  
    Ck = apriori – gen(Lk-1); Compute new candidates  
    for all Transaktionen t ∈ D do begin  
        Ct = subset(Ck, t); Compute all candidates in t  
        for all Candidates c ∈ Ct do  
            c.count ++;  
    end  
    Lk = {c ∈ Ck | c.count ≥ minsup}  
end  
return: ∪k Lk
```

- Runtime behavior of the a priori:
 - good for small and medium data volumes
 - Bad for very large amounts of data
- Solution through modifications:
 - AprioriTID
 - AprioriHybrid

1.5 Classification

- Classification and concept learning
- ID3
- C5.0

Most common business analytics jobs

Problem	Business Perspective	Techniques
Find Clusters/Outliers	<ul style="list-style-type: none"> Are there different types of users Can we put different products together into distinct/different groups? 	<ul style="list-style-type: none"> Clustering Outlier-Analysis
Find Relationships	<ul style="list-style-type: none"> If a customer buys product A, what does he buy next? Which product sets belong together? 	<ul style="list-style-type: none"> Association Analysis
Predict Classes	<ul style="list-style-type: none"> Is this customer solvent or not? Will this customer send back this shipping or not? 	<ul style="list-style-type: none"> Decision Trees Logistic Regression Support-Vector Machines
Predict Values	<ul style="list-style-type: none"> Does a new label increase the? Is there a relationship between Sales and Commercials? 	<ul style="list-style-type: none"> Regression Support-Vector Machines
Predict Developments	<ul style="list-style-type: none"> How will the value of our products develop? What future developments are likely? 	<ul style="list-style-type: none"> Time-Series Forecasting



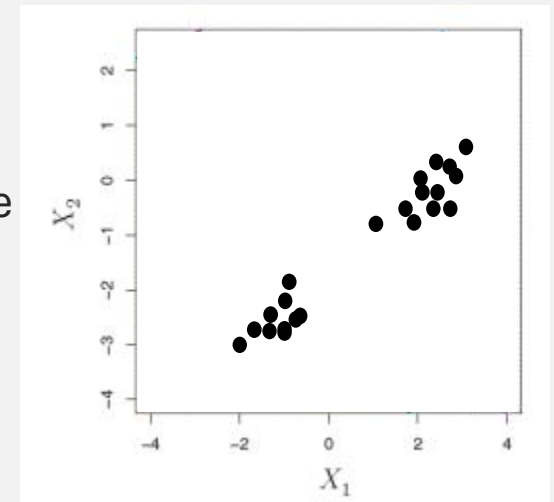
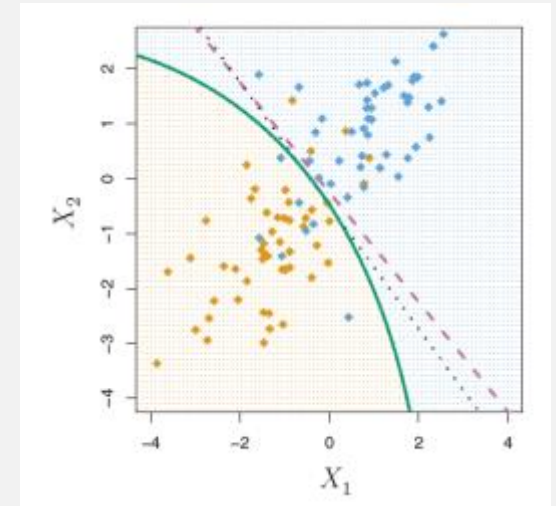
There will be further advanced techniques we will discuss in „Advanced Analytics with R“, another lecture covering topics like e.g. ANN, FP Growth techniques, Expectation Maximization etc.

Classification

- We have a set of observations with different attributes and known class labels (the attribute data scientist want to predict is termed the class attribute)
- We want to built a model to predict the class attribute for new observations with the same attributes (so we know all other attributes except the class attribute)
- The most popular tools for that are decision trees

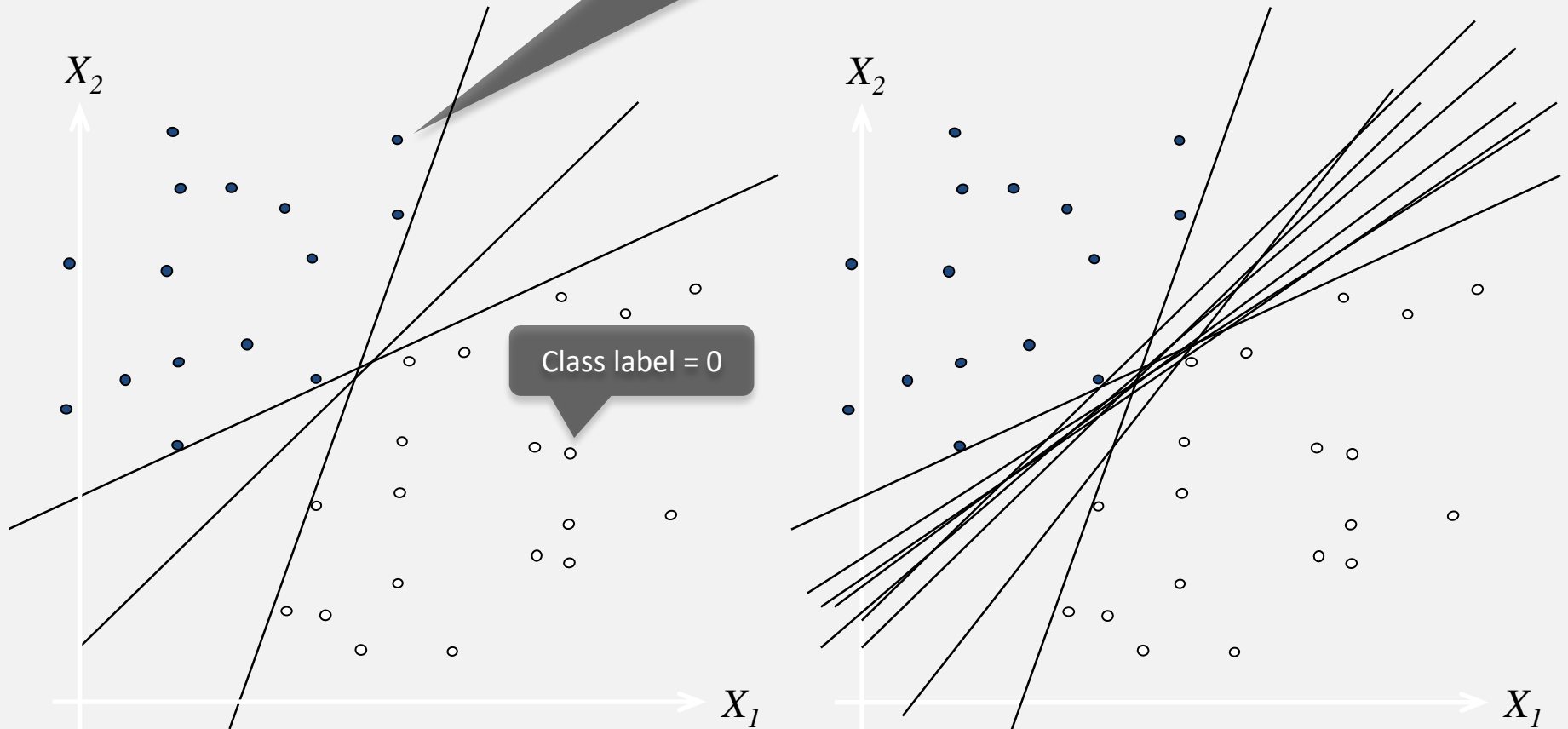
Clustering vs. Classification

- Objects are characterized by features
- Classification:
 - Have labels for some points (training data)
 - Want a “rule” to assign labels to new points (test data)
 - Distance between objects “doesn’t matter”
 - Supervised learning
- Clustering:
 - No labels
 - Group points based on, e.g., how “near” they are to one another
 - Identify structure in data... and reduce data
 - Unsupervised learning



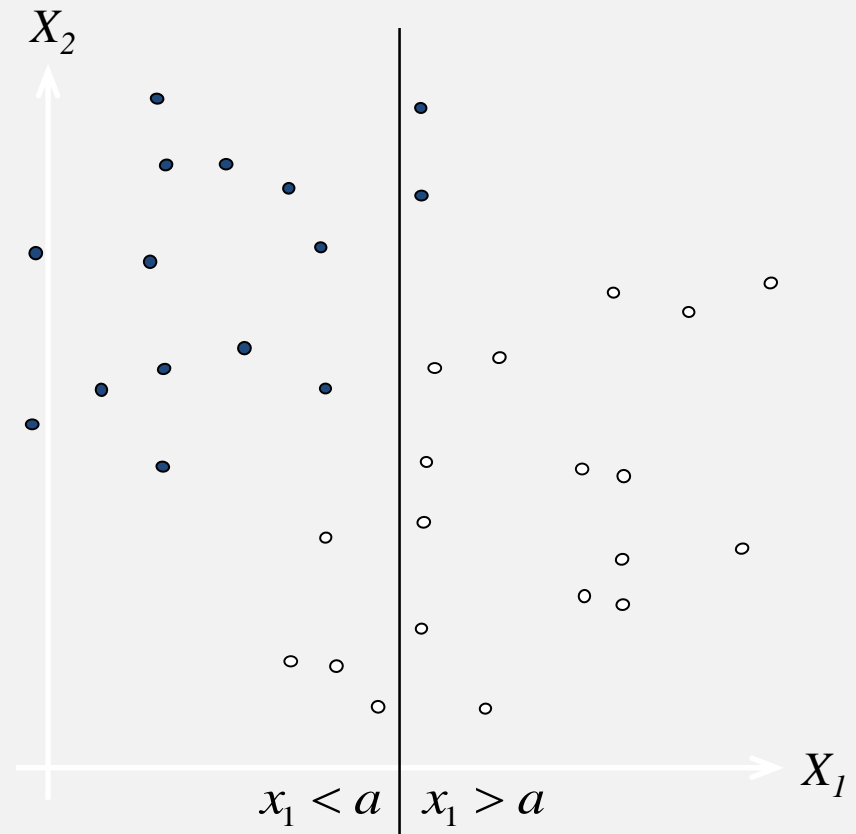
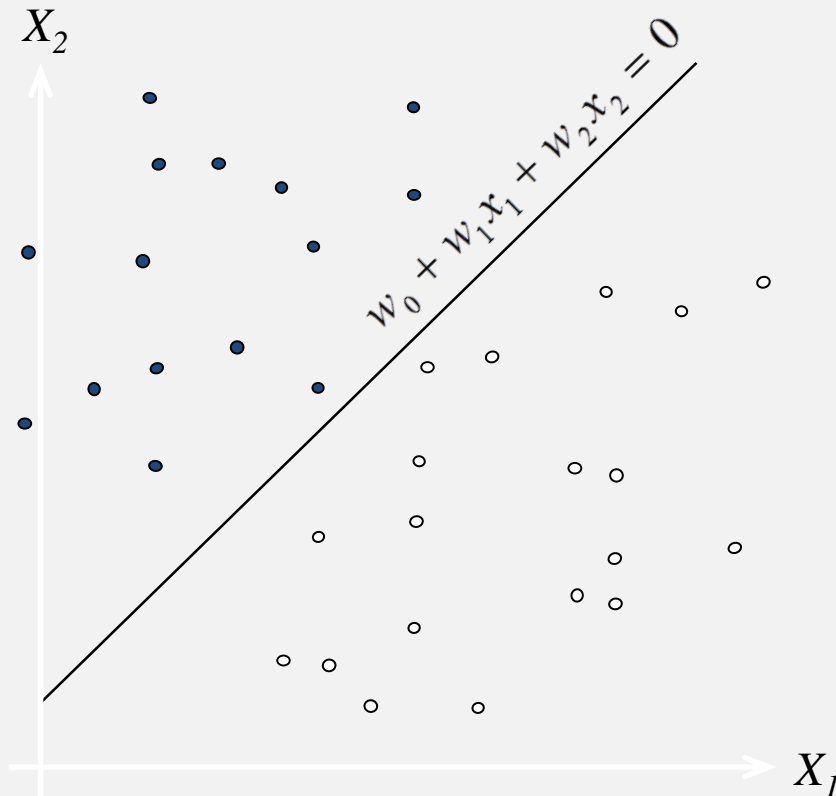
Linear Classifiers

Class label = 1, e.g., purchased a target item after a campaign



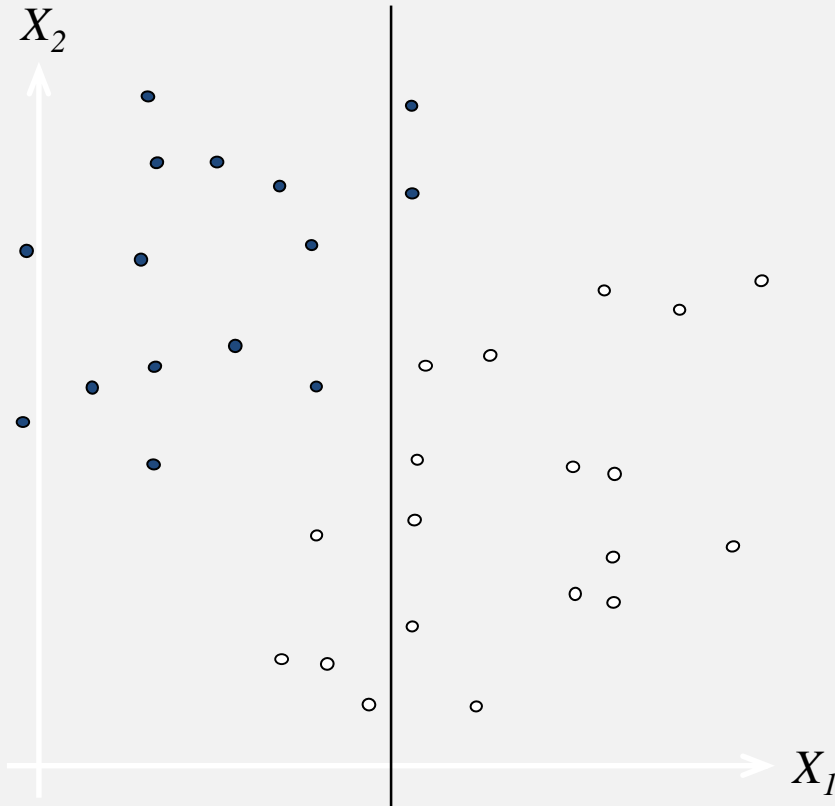
- Any of these would be fine ...
- ... but which is the best?

Linear Classifiers and Decision Trees



- What about this?
- Advantage? Disadvantage?

Entropy as a measure of Impurity



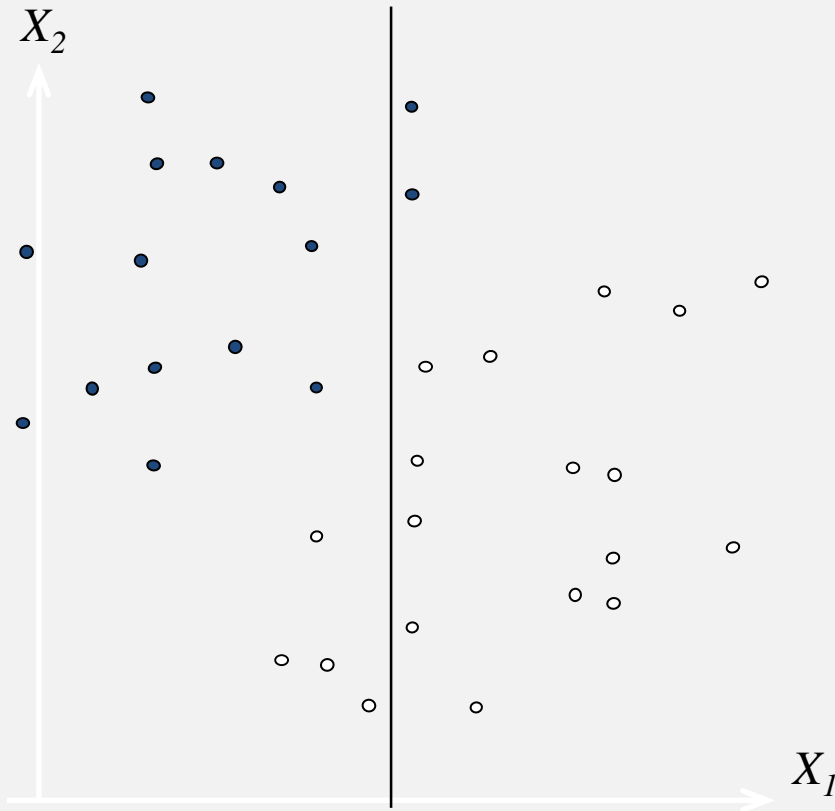
- **Wanted:** a function that describes the “impurity” of a set of data points

- **Solution:** *Entropy*

$$H(S) = - \sum_{i=1}^n p_i \cdot \log_2 p_i$$

Hier: $H(S) = -p_1 \log_2(p_1) - p_2 \log_2(p_2)$

Entropy and Information Gain



Entire population (34 instances)

“Parent entropy“:

$$-\left(\frac{15}{34}\log_2\frac{15}{34}\right)-\left(\frac{19}{34}\log_2\frac{19}{34}\right)=0.989$$

“Child entropy“ of left side (17 instances):

$$-\left(\frac{13}{17}\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\log_2\frac{4}{17}\right)=0.787$$

“Child entropy“ of right side (17 instances):

$$-\left(\frac{2}{17}\log_2\frac{2}{17}\right)-\left(\frac{15}{17}\log_2\frac{15}{17}\right)=0.523$$

Entropy of children:

$$\left(\frac{17}{34}0.787\right)+\left(\frac{17}{34}0.523\right)=0.655$$

Information Gain:

$$0.989 - 0.655 = 0.334$$



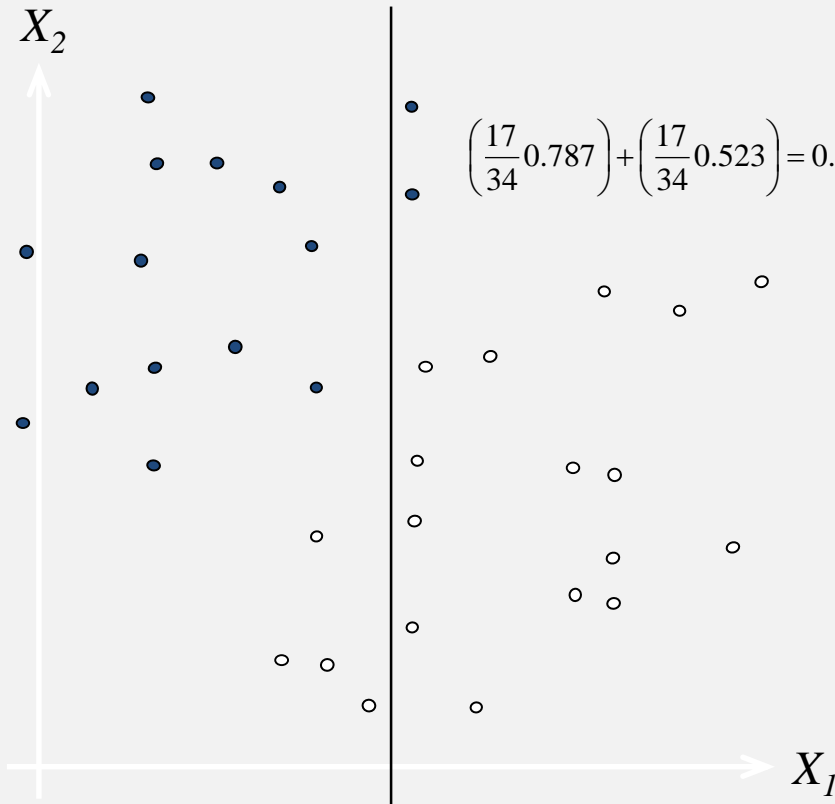
Remember *Lecture 4 - Business Data Preparation*: Information gain tells us how good the given split on an attribute is

Information Gain

Entire population (34 instances):

$$-\left(\frac{15}{34}\log_2\frac{15}{34}\right)-\left(\frac{19}{34}\log_2\frac{19}{34}\right)=0.989$$

$$\left(\frac{12}{34}0.811\right)+\left(\frac{22}{34}0.845\right)=0.833$$



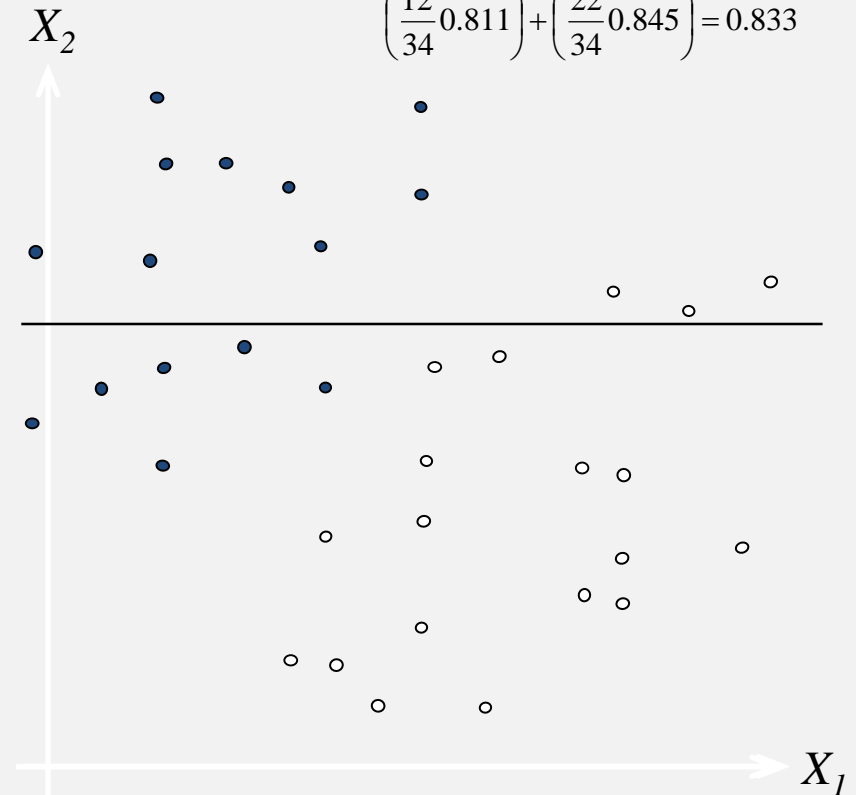
Entropy left
(17 instances):

$$-\left(\frac{13}{17}\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\log_2\frac{4}{17}\right)=0.787$$

Entropy right
(17 instances):

$$-\left(\frac{2}{17}\log_2\frac{2}{17}\right)-\left(\frac{15}{17}\log_2\frac{15}{17}\right)=0.523$$

$$\left(\frac{17}{34}0.787\right)+\left(\frac{17}{34}0.523\right)=0.655$$



Entropy above
(12 instances):

$$-\left(\frac{9}{12}\log_2\frac{9}{12}\right)-\left(\frac{3}{12}\log_2\frac{3}{12}\right)=0.811$$

Entropy below
(22 instances):

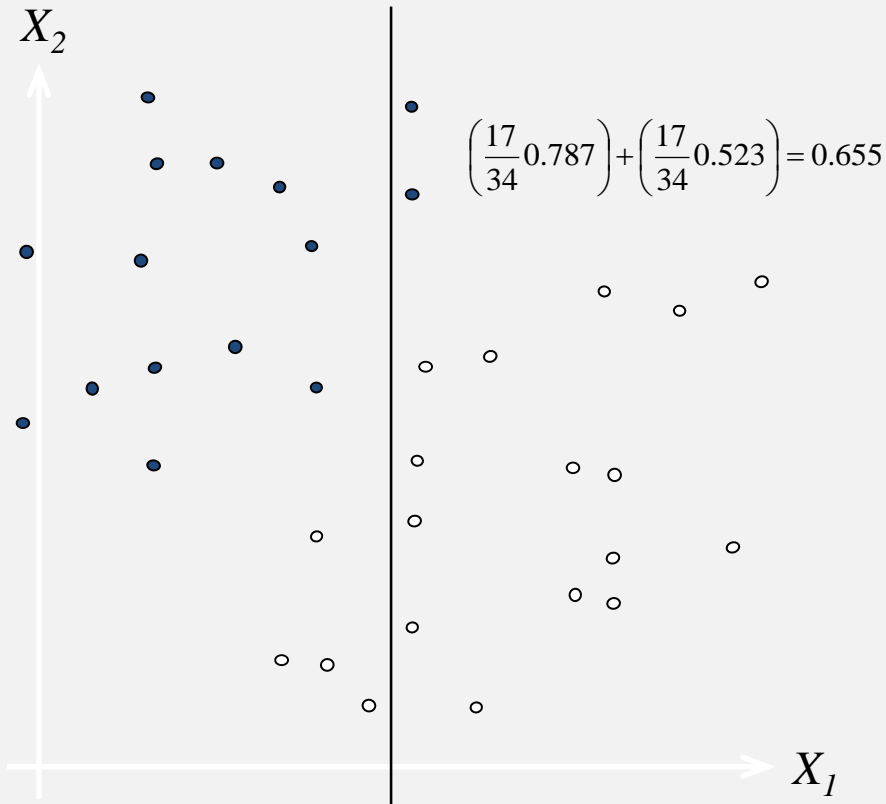
$$-\left(\frac{6}{22}\log_2\frac{6}{22}\right)-\left(\frac{16}{22}\log_2\frac{16}{22}\right)=0.845$$

Information Gain

Entire population (34 instances):

$$-\left(\frac{15}{34}\log_2\frac{15}{34}\right)-\left(\frac{19}{34}\log_2\frac{19}{34}\right)=0.989$$

$$\left(\frac{19}{34}0\right)+\left(\frac{25}{34}0.795\right)=0.585$$



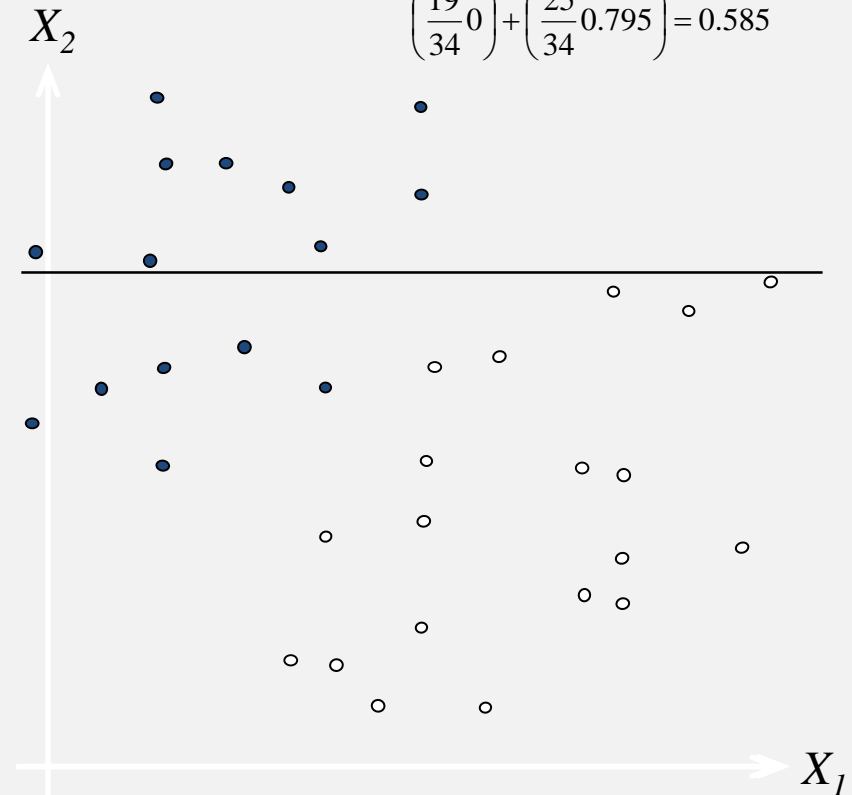
Entropy left
(17 instances):

$$-\left(\frac{13}{17}\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\log_2\frac{4}{17}\right)=0.787$$

Entropy right
(17 instances):

$$-\left(\frac{2}{17}\log_2\frac{2}{17}\right)-\left(\frac{15}{17}\log_2\frac{15}{17}\right)=0.523$$

$$\left(\frac{17}{34}0.787\right)+\left(\frac{17}{34}0.523\right)=0.655$$



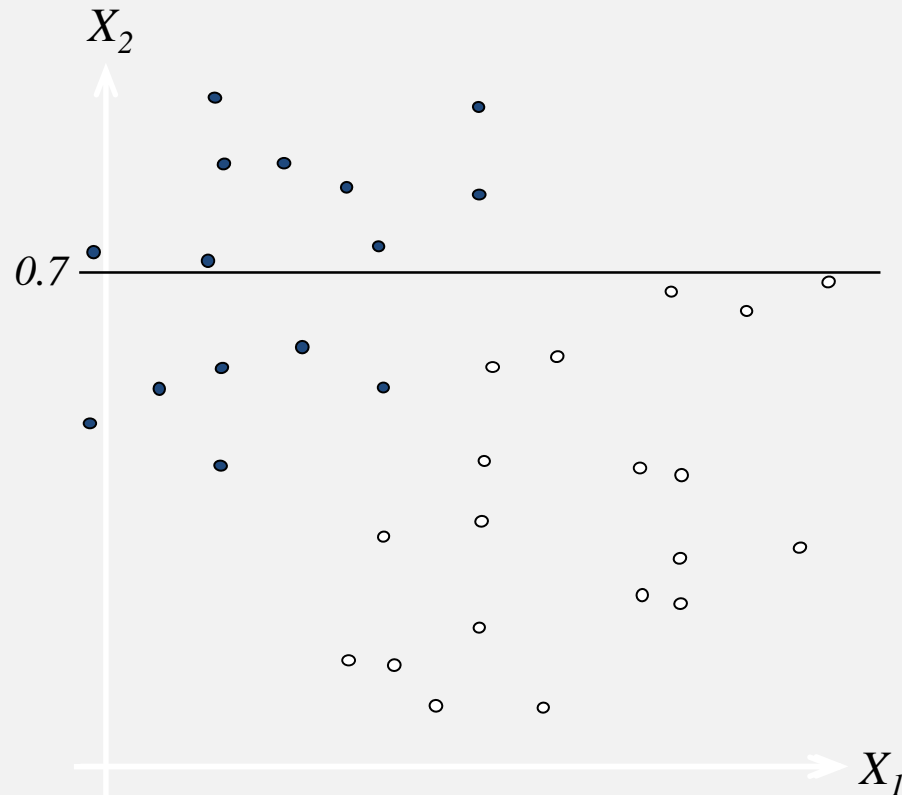
Entropy above
(9 instances):

$$-\left(\frac{9}{9}\log_2\frac{9}{9}\right)-\left(\frac{0}{9}\log_2\frac{0}{9}\right)=0$$

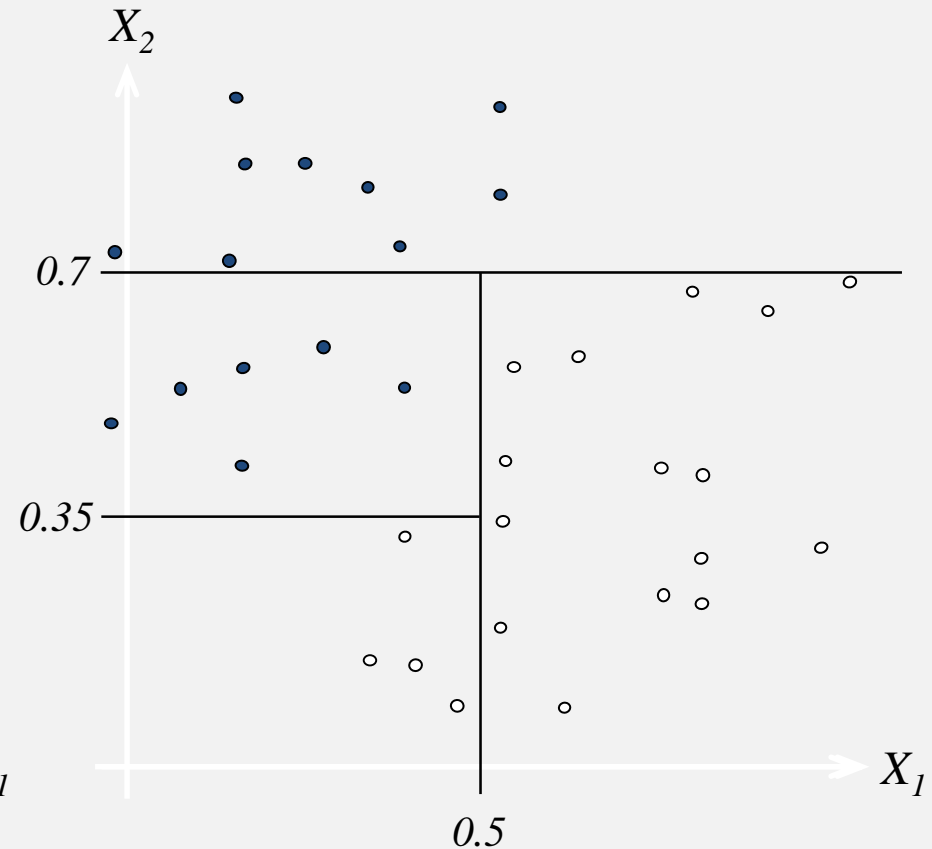
Entropy below
(25 instances):

$$-\left(\frac{6}{25}\log_2\frac{6}{25}\right)-\left(\frac{19}{25}\log_2\frac{19}{25}\right)=0.795$$

Information Gain

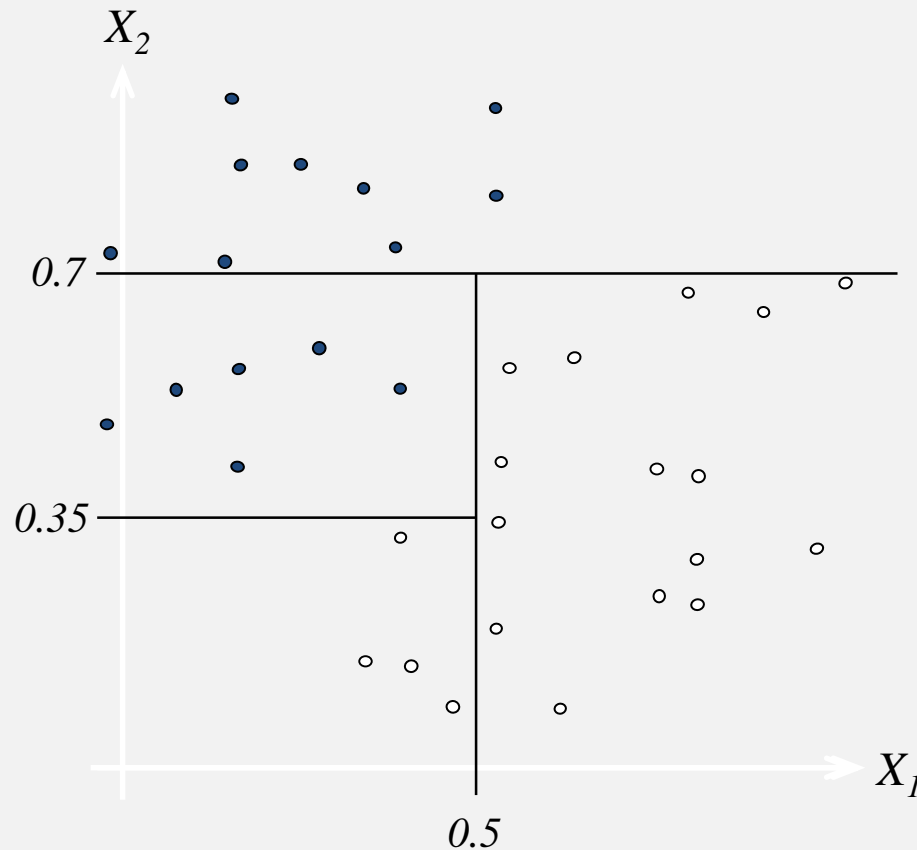


- Select the variable (here X_2) and split (here at 0.7) that yields the highest information gain (best discriminator)



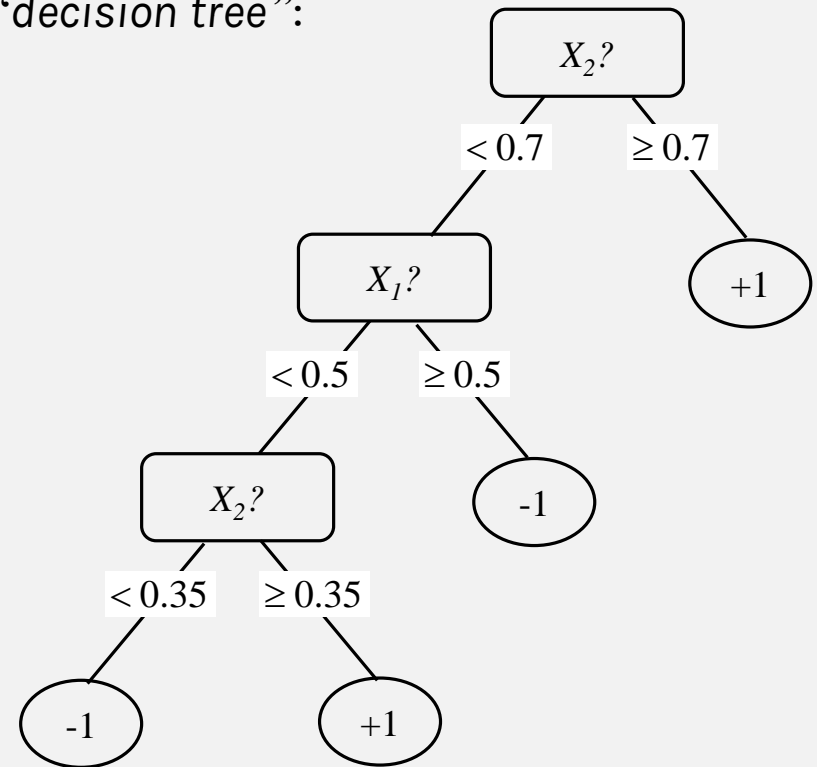
- Do this iteratively until no more impurity exists (the set of points is fully classified)

Decision Trees



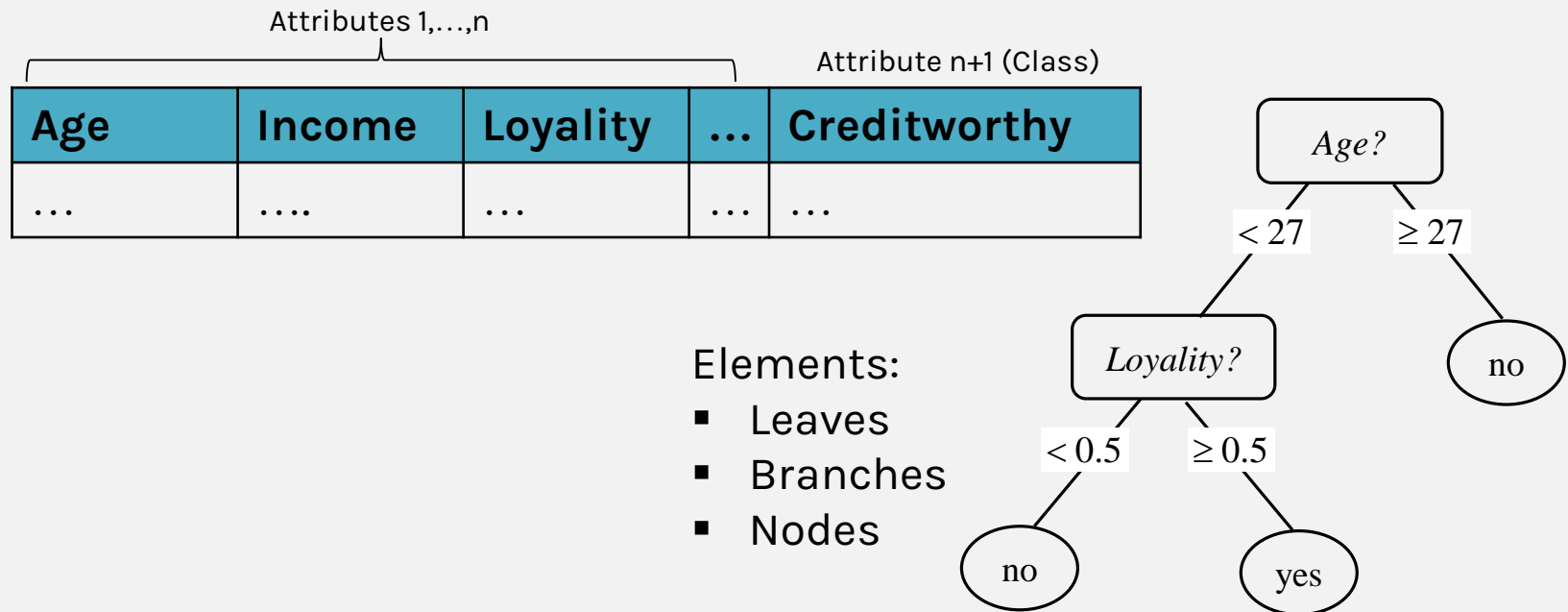
- denotes +1
- denotes -1

- This iterative process of splitting can be expressed as a “decision tree”:



Concept of decision trees

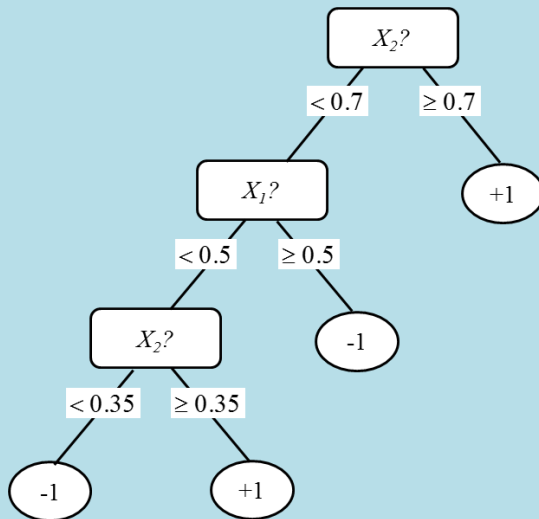
- **Objective:** Item has several attributes, you want to predict the (n+1)-th using the values of n attributes
- **Basis for prediction:** quantity of tuples (training quantity) for which all n+1 values are known.
- Example:



Concept Learning System (CLS)

Hunt et al. (1966)

- Built a model to predict or classify future observations based on a set of decision rules



- All decision tree algorithms are based on Hunt's fundamental algorithm of concept learning
- The algorithm embodies a method used by humans when learning simple concepts, namely finding key distinguishing features between two categories, represented by positive and negative (Training) examples
- Based on a divide and conquer strategy

Hunt's Algorithm

1. If all data objects belong to the same class, you form a leaf in the decision tree.
2. Otherwise:
 - a) An attribute is selected,
 - b) The number of data objects is partitioned according to the attribute values that occur for this attribute, which represent each successive node and
 - c) The algorithm is recursively applied to these successor nodes and the Remaining attributes

- A decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute
- Each branch represents an outcome of the test
- Each leaf node (terminal node) holds a class label

Classification with Decision Trees

Decision Trees classify instances

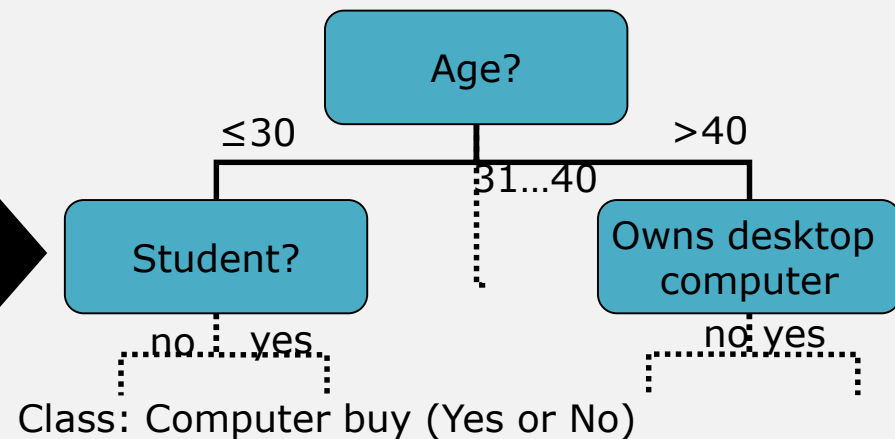
- Sorting them down tree to leaf node containing class (value)
- Based on attributes of instances
- Branch for each value

Example: Decision Tree indicating whether or not customer buys computer

How to build such Decision Trees from sample data?

Class: Computer Buy	Age	Student	Owns Desktop
Yes	32	Yes	No
No	12	No	No
...
...
...

Classifi-
cation
algorithm



Training Data

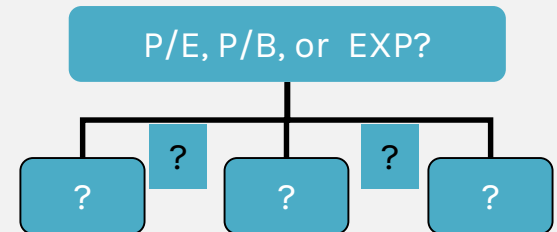
Instances presented as attribute-value pairs

Observation	Investment decision	Price-earnings ratio (P/E)*	Price-book ratio (P/B)*	Expert opinion (EXP)
1	Buy	3	3	Good
2	Sell	1	1	Bad
3	Buy	3	2	Good
4	Sell	1	2	OK
5	Sell	1	1	OK
6	Sell	2	1	OK
7	Buy	3	3	Good
8	Buy	3	2	Good
9	Buy	2	3	OK
10	Sell	2	2	OK
11	Sell	1	2	OK
12	Sell	2	1	Bad
13	Buy	2	3	Good
14	Sell	1	2	Bad

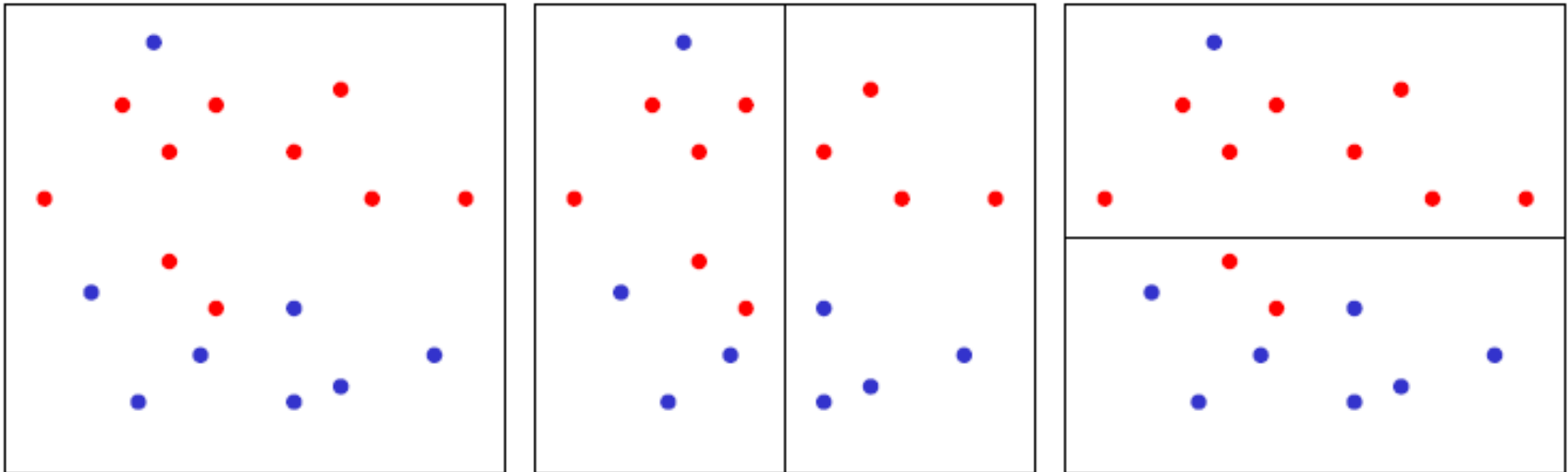
* Normalized to a 1-N ranking

Built decision trees with ID3

- ID3 classification algorithm generates a decision tree from a fixed set of examples (observations)
- Resulting tree is used to classify future samples



How to decide which attribute to split on (1st, 2nd,...)?



[Source: P. Cimiano]

Built decision trees with ID3

- **For each attribute that has not already been used**
 - Calculate the information gain that results from splitting on that attribute
 - Split on the attribute that gives the greatest information gain
- Attributes are chosen repeatedly in this way until a complete decision tree that classifies every input is obtained

How to use ID3 on existing datasets?

- Precise definition for information gain...
- Example

ID3: Entropy & Information Gain

Observation	Expected return
1	Buy
2	Sell
3	Buy
4	Sell
5	Sell
6	Sell
7	Buy
8	Buy
9	Buy
10	Sell
11	Sell
12	Sell
13	Buy
14	Sell

ID3 develops Decision Trees on the basis on two criteria:

1. **Entropy** (Overall level of uncertainty)
2. **Information Gain** (Information being the most useful for classification)

$$H(S) = - \sum_{i=1}^n p_i \cdot \log_2 p_i$$

$$\begin{aligned} &= -p_{\text{Sell}} \log_2(p_{\text{Sell}}) - p_{\text{Buy}} \log_2(p_{\text{Buy}}) \\ &= -(8/14) \log_2(8/14) - (6/14) \log_2(6/14) \\ &= \underline{\underline{0,98522814}} \end{aligned}$$

ID3: Entropy & Information Gain (2)

Information gain is computed for each attribute A : $Gain(A) = H(S) - H(A)$

Observation	Investment Decision	Expert opinion (EXP)
1	Buy	Good
2	Sell	Bad
3	Buy	Good
4	Sell	OK
5	Sell	OK
6	Sell	OK
7	Buy	Good
8	Buy	Good
9	Buy	OK
10	Sell	OK
11	Sell	OK
12	Sell	Bad
13	Buy	Good
14	Sell	Bad

$$Gain(S, A_{EXP}) = Entropy(S)$$

$$- (3/14) Entropy(S_{Bad})$$

$$- (6/14) Entropy(S_{OK})$$

$$- (5/14) Entropy(S_{Good})$$

$$Entropy(S_{Bad}) = -(3/3) \log_2(3/3) - 0 \log_2 0 = 0$$

$$Entropy(S_{OK}) = -(5/6) \log_2(5/6) - (1/6) \log_2(1/6)$$

$$\approx 0.65002$$

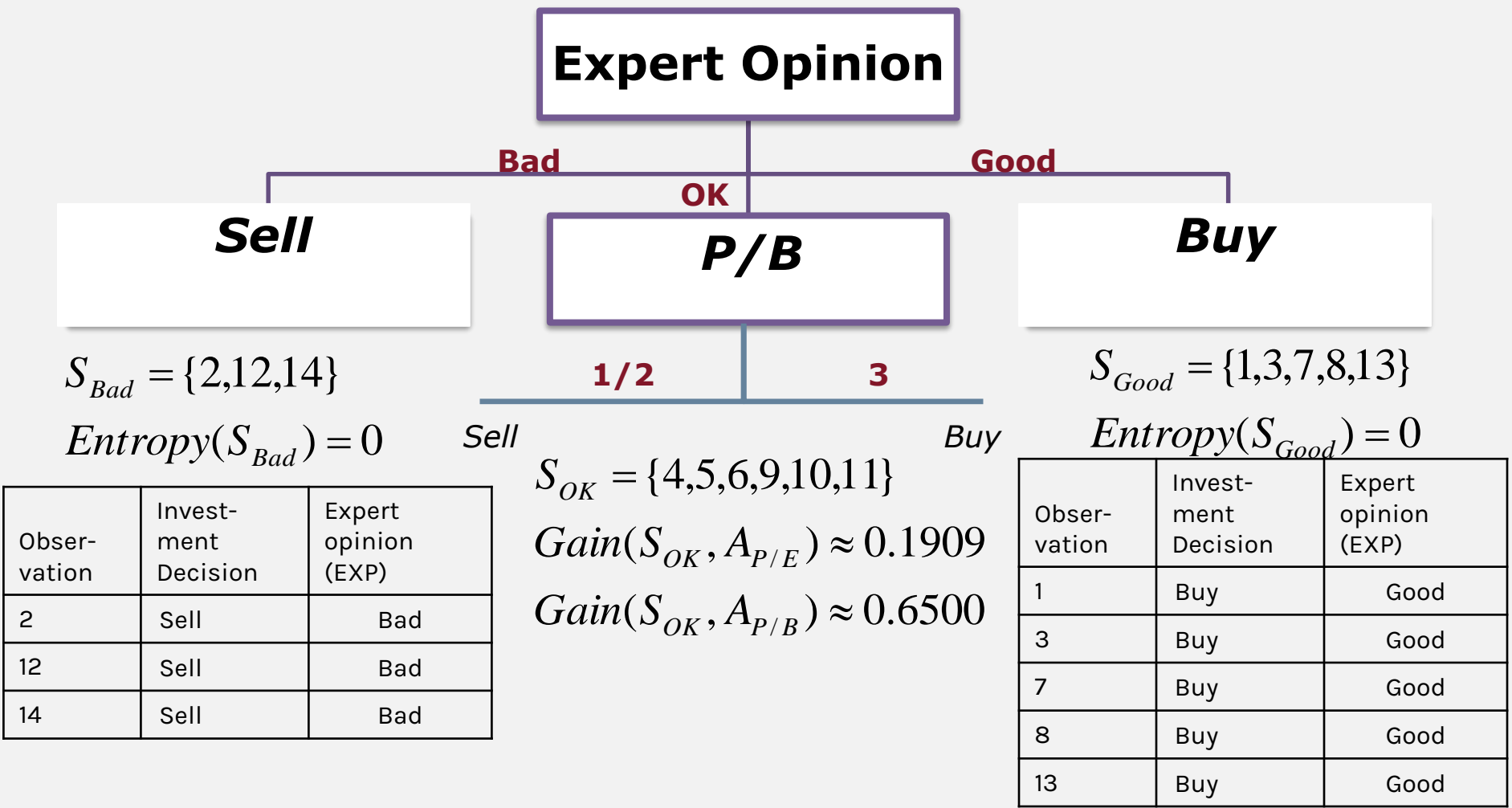
$$Entropy(S_{Good}) = -0 \log_2 0 - (5/5) \log_2(5/5) = 0$$

$$Gain(S, A_{EXP}) \approx 0.98522 - 0 - 6/14 \cdot 0.65002 - 0$$

$$\approx \underline{\underline{0.7066471}}$$

$$Gain(S, A_{P/E}) \approx 0.63846007$$

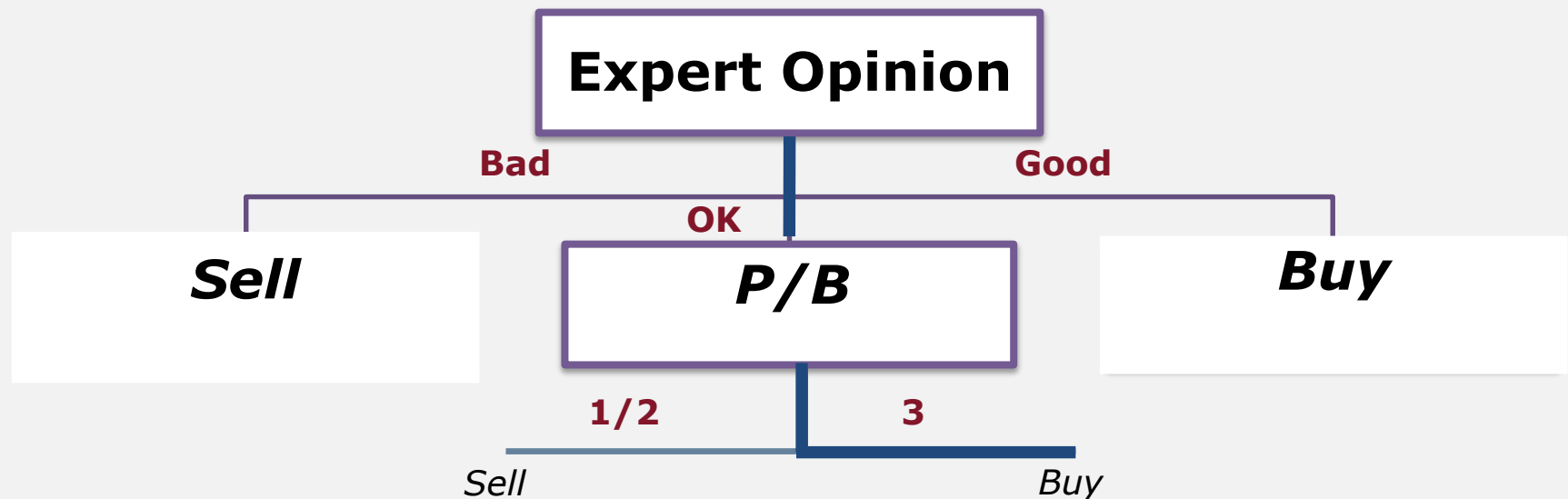
$$Gain(S, A_{P/B}) \approx 0.59167278$$



Model Application

- Given the developed model, we can classify instances, e.g.:

Observation	Investment decision	Price-earnings ratio (P/E)*	Price-book ratio (P/B)*	Expert opinion (EXP)
1	Buy	3	3	OK



- With a larger testing set, we could test the model's classification accuracy.



The `rpart` function is part of the package `rpart`.

`rpart()`

```
fit = rpart(formula = class ~ att1 + att2 +..., data = x, parms =  
list(prior = c(.65,.35), split = "information"))
```

Parameters

formula	Formula, with a response but no interaction terms
x (Input)	Numeric matrix or data frame that can be coerced to such a matrix (data frame with all numeric columns)
parms	<p>Optional parameters for the splitting function, multiple parameters with a list. For instance:</p> <ul style="list-style-type: none">▪ vector of prior probabilities (component prior), the loss matrix (component loss) or the splitting index (component split). The priors must be positive and sum to 1. The loss matrix must have zeros on the diagonal and positive off-diagonal elements.▪ The splitting index can be gini or information

Output of an ID3 model

```
library(rpart)
fit = rpart(formula = mpg ~ ., data = mtcars, parms = list(split = "information"))
summary(fit)
```

Call:
rpart(formula = mpg ~ ., data = mtcars, parms = list(split = "information")) n= 32

	CP	nsplit	rel error	xerror	xstd
1	0.64312523	0	1.0000000	1.0886799	0.2546997
2	0.09748407	1	0.3568748	0.6649430	0.1691361
3	0.01000000	2	0.2593907	0.5133424	0.1132490

The parametrization of the decision tree. The „.“ in the formula stands for „all the remaining variables“

Variable importance

cyl	disp	hp	wt	qsec	vs	carb
20	20	19	16	12	11	1

Individual importance of the features of your dataset

Predictive performance of the model to decide where to prune it: Resubstitution error rate and cross-validated error rate

Node number 1: 32 observations, complexity param=0.6431252
mean=20.09062, MSE=35.18897
left son=2 (21 obs) right son=3 (11 obs)
Primary splits:

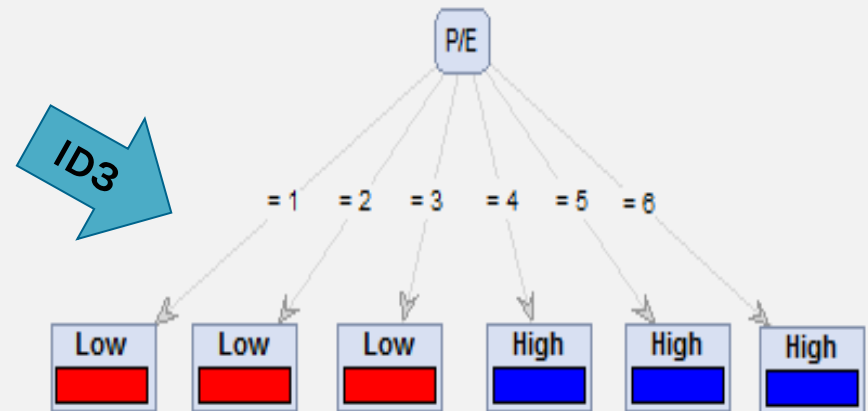
cyl < 5	to the right, improve=0.6431252, (0 missing)
wt < 2.3925	to the right, improve=0.6356630, (0 missing)
disp < 163.8	to the right, improve=0.6130502, (0 missing)
hp < 118	to the right, improve=0.6010712, (0 missing)

Your decision tree

ID3 Overfitting

- ID3 Disadvantage: Tends to prefer splits that result in large number of partitions, each being small but pure.

#	Expected return	(P/E)	(P/B)	(EXP)
1	High	5	3	Good
2	Low	1	1	Bad
3	High	6	2	Good
4	Low	2	2	OK
5	Low	3	1	OK
6	High	4	1	OK



- P/E provides maximum information gain, but...
...is the sensible with regard to model developing (generalization)?

Father of decision trees: Ross Quinlan



Ross Quinlan

Pioneer of decision tree research

Image source: ↗[Ross Quinlan](#) (2018)

- Computer science researcher in data mining and decision theory at university of Sydney
- Very famous paper “Quinlan, J. R. (1986). *Induction of decision trees*. *Machine learning*, 1(1), 81-106.”
- Founded the company *RuleQuest Research* in 1997, which offers data analytics solutions
- Building on the ID3 he create the C4.5 and the C5 algorithm,

C4.5: Central Concepts

- C4.5 develops decision trees on the basis of two criteria:
 - **InformationGain** (Information being the most useful for classification)
 - **SplitInfo** (Information how an attribute splits the data)
- On the basis of these two criteria, C4.5 calculates and uses *GainRatio* instead of *InformationGain* (as *ID3*)
- Improvements: can handle discrete and continuous attributes, can handle missing attribute values, attributes with differing costs, improved pruning of trees (replacing irrelevant branches with leaf nodes).

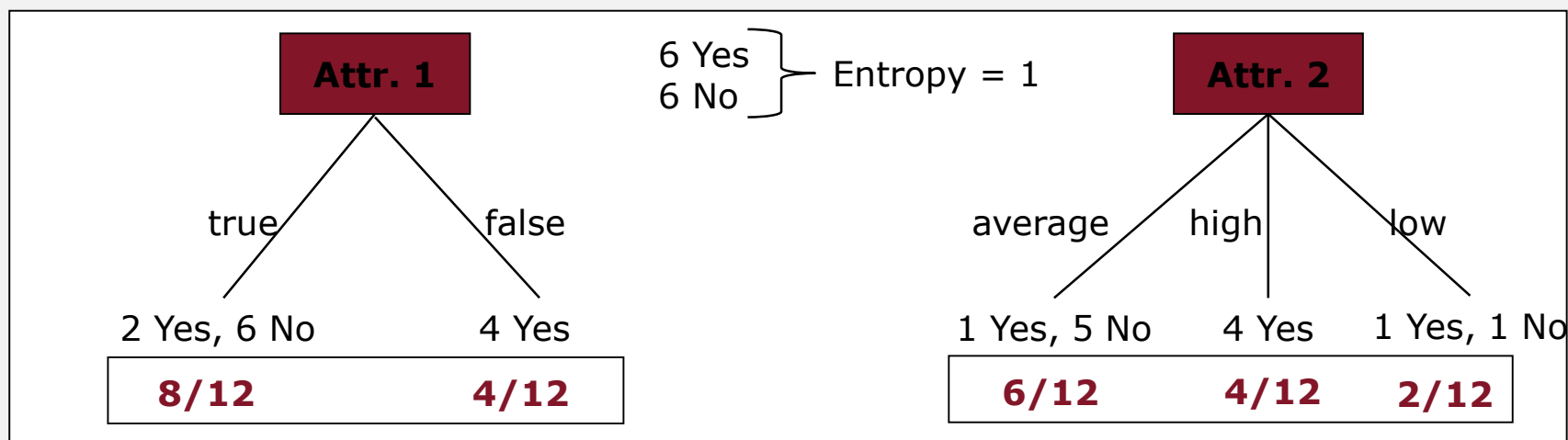
$$\text{GainRatio}_{split} = \frac{\text{InfoGain}_{split}}{\text{SplitInfo}} \quad \text{SplitInfo} = -\sum_{i=1}^k \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

- Adjusts *Information Gain* (InfoGain_{split}) by entropy of partitioning (SplitInfo)
- Higher entropy partitioning (large number of small partitions) is penalized
- Developed to overcome disadvantage of *Information Gain* approach of ID3

[Quinlan 1992]

Normalizing Information Gain: ID3 vs. C4.5

- C4.5 fixes problem of ID3 by normalizing *Information Gain* of each candidate attribute, taking into account number of branches



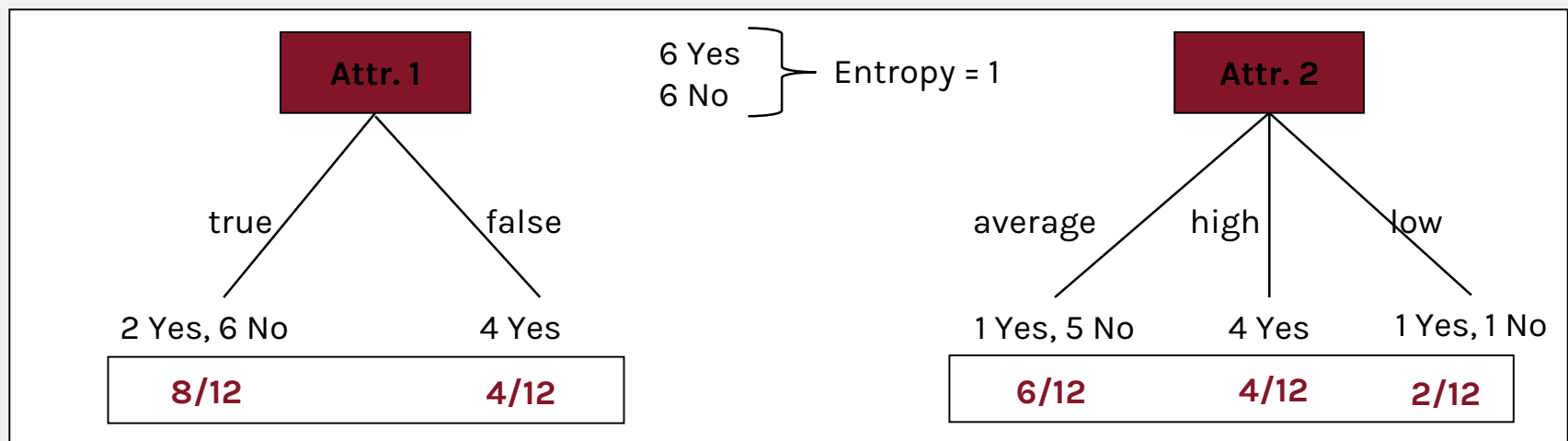
- ID3:

$$\text{InfoGain}(\text{Attr.1}) = 1 - 8/12 \left[\left(-2/8 \log_2(2/8) - 6/8 \log_2(6/8) \right) \right] + 0 = 0.459$$

$$\begin{aligned} \text{InfoGain}(\text{Attr.2}) &= 1 - 6/12 \left[\left(-1/6 \log_2(1/6) - 5/6 \log_2(5/6) \right) \right] + 0 \\ &\quad - 2/12 \left[\left(-1/2 \log_2(1/2) - 1/2 \log_2(1/2) \right) \right] = 0.508 \end{aligned}$$

↪ better tree?

Normalizing Information Gain: ID3 vs. C4.5



$$SplitInfo(Attr.1) = -4/12 \log_2(4/12) - 8/12 \log_2(8/12) = 0.918$$

$$InfoGain(Attr.1) = 0.459$$

$$GainRatio(Attr.1) = 0.459 / 0.918 = 0.5$$

$$SplitInfo(Attr.2) = -6/12 \log_2(6/12) - 4/12 \log_2(4/12) - 2/12 \log_2(2/12) = 1.459$$

$$InfoGain(Attr.2) = 0.508$$

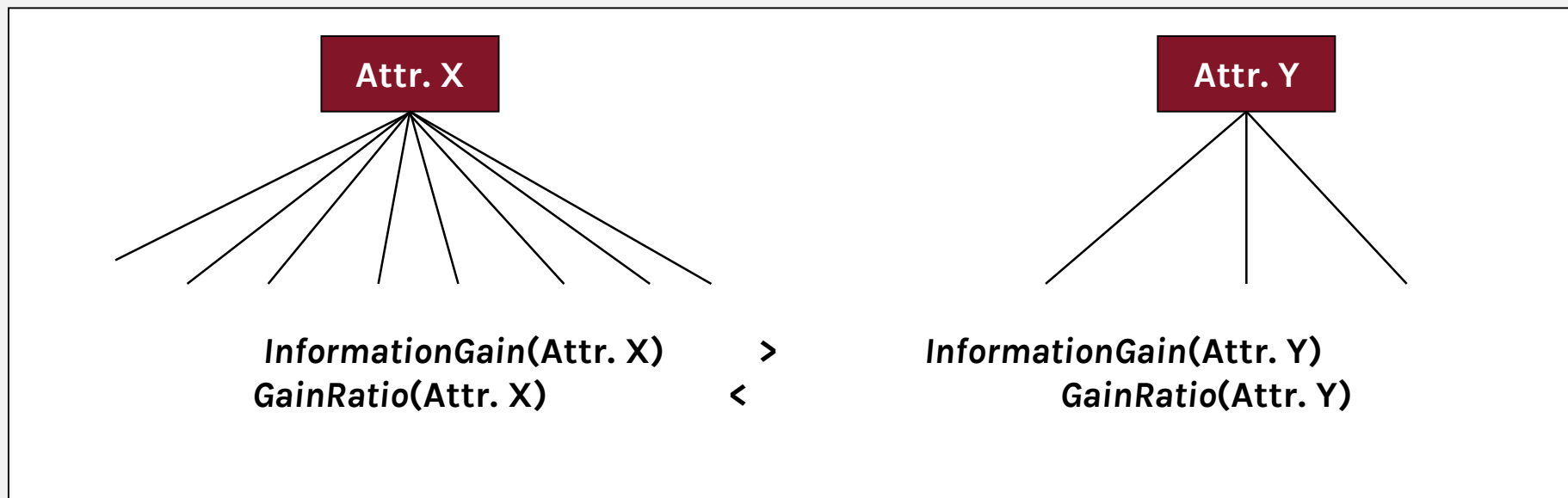
$$GainRatio(Attr.2) = 0.508 / 1.459 = 0.348$$

More branches penalized

C4.5 selects
Attr. 1

Normalizing Information Gain: ID3 vs. C4.5

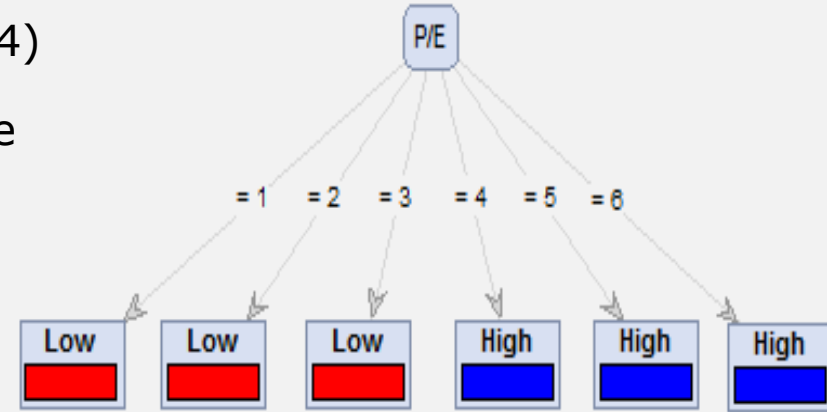
- C4.5 normalizes Information Gain introduced by ID3, i.e. it penalizes attributes that would result in more branches.
- Experimentally, *GainRatio* seems to be better than *InformationGain* as a measure of the usefulness of attribute to classification problem.



Numeric Attributes

Standard method: binary splits (e.g. $P/E < 4$)

- Unlike nominal attributes, every attribute has many possible split points



Basic approach:

- Place split points halfway between values

Extended approach:

- Evaluate for every possible split point of attribute
- Choose “best” split point
- Best split point creates two sets with maximal measure (e.g. *GainRatio*)

→ **Computationally demanding!**

Example: Weather Data – Nominal Values

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	Normal	False	Yes
...

If outlook = sunny and humidity = high then play = no

If outlook = rainy and windy = true then play = no

If outlook = overcast then play = yes

If humidity = normal then play = yes

If none of the above then play = yes

Example: Weather Data – Numeric

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	75	80	False	Yes
...

`If outlook = sunny and humidity > 83 then play = no`

`If outlook = rainy and windy = true then play = no`

`If outlook = overcast then play = yes`

`If humidity < 85 then play = yes`

`If none of the above then play = yes`

Example: Weather Data - Numeric

Split on temperature attribute:

value	64	65	68	69	70	71	72	73	75	75	80	81	83	85
class	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

Approaches:

1. Place split points halfway between values, i.e.

temperature < 73 : yes/4, no/3

temperature ≥ 73 : yes/5, no/2

or

2. Evaluate all split points with regard to *GainRatio*, e.g.

temperature < 71.5 : yes/4, no/2

temperature ≥ 71.5 : yes/5, no/3


Choose split that results highest *GainRatio*

⇒ **13 calculations necessary in this case?**

Example: Weather Data - Numeric

- Sort instances by the values of numeric attribute
- *GainRatio* only needs to be evaluated between points of different classes (Fayyad & Irani, 1992)

value	64	65	68	69	70	71	72	72	75	75	80	81	83	85
class	Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No



- Potential optimal breakpoints, i.e. 7 calculations necessary in this case
- Breakpoints between values of the same class cannot be optimal.

Binary vs. Multi-way Splits

Splitting (**multi-way**) on a nominal attribute **exhausts all information** in that attribute

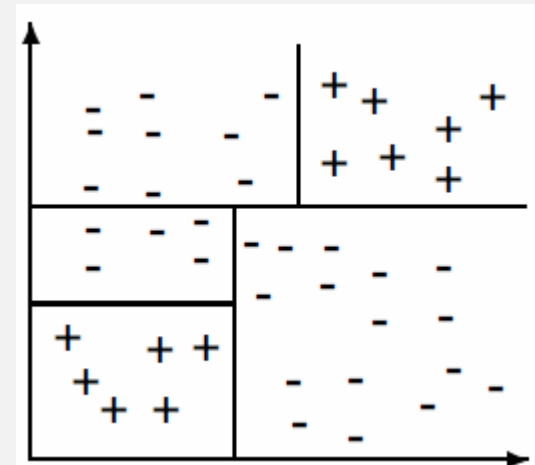
- Nominal attribute is tested (at most) once on any path in tree

Not so for **binary splits** on numeric attributes

- Numeric attribute may be tested several times along path in tree

C4.5 approach to numeric values is often useful but not in all cases

- In some cases, a single split will not increase information



Missing Values

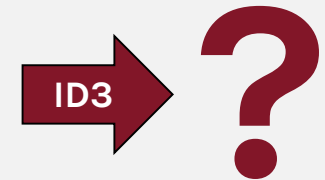
Missing attribute values are common in data:

- Values may not always be available
- Values may not have been considered important at time of entry
- Values may not have been recorded due to misunderstanding, or because of equipment malfunctions
- Data that were inconsistent with other recorded data may have been deleted

[Han & Kamber 2006]

ID3 can not handle cases where some values of attributes are unknown

Application	Attr.1
Yes	true
Yes	?
No	true



Missing Values

- C4.5 allows missing values to have the form “?”
- Basic approach to calculate *InformationGain* works as before, unknown values are not included in calculations.

S_{all} = Entire sample including missing values

S_{sub} = Subsample, where all values of attributes are known

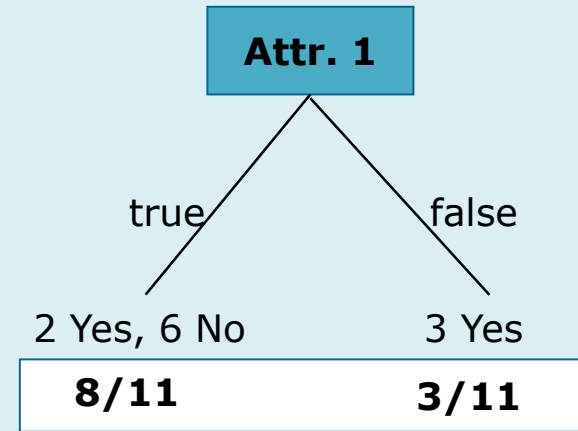
1. Calculate *Entropy* for S_{sub}

2. Calculate
$$InfoGain(S_{all}, A) = \frac{n(S_{sub})}{n(S_{all})} \cdot InfoGain(S_{sub}, A)$$

3. Calculate *SplitInfo* by treating unknown values as additional class

Missing Values

Application	Attr.1	Attr.2
Yes	true	
Yes	true	
No	true	
Yes	false	
No	true	
No	true	
Yes	?	
No	true	
Yes	false	
No	true	
Yes	false	
No	true	



$$Entropy(S_{Sub}) = -8/11 \cdot \log_2 8/11 - 3/11 \cdot \log_2 3/11$$

$$InfoGain(S_{Sub}, Attr.1) = Entropy(S_{Sub}) - 8/11 Entropy(S_{Sub}, A1 = true) - 3/11 Entropy(S_{Sub}, A1 = false)$$

$$InfoGain(S_{all}, Attr.1) = 11/12 \cdot InfoGain(S_{Sub}, Attr.1)$$

$$SplitInfo = -8/12 \log_2 8/12 - 3/12 \log_2 3/12 - 1/12 \log_2 1/12$$

$$GainRatio(S_{All}, Attr.1) = \frac{InfoGain(S_{All}, Attr.1)}{SplitInfo}$$

Pruning

- When decision tree is built, many of branches will reflect anomalies in training data due to noise or outliers
- Pruning goal: Prevent overfitting

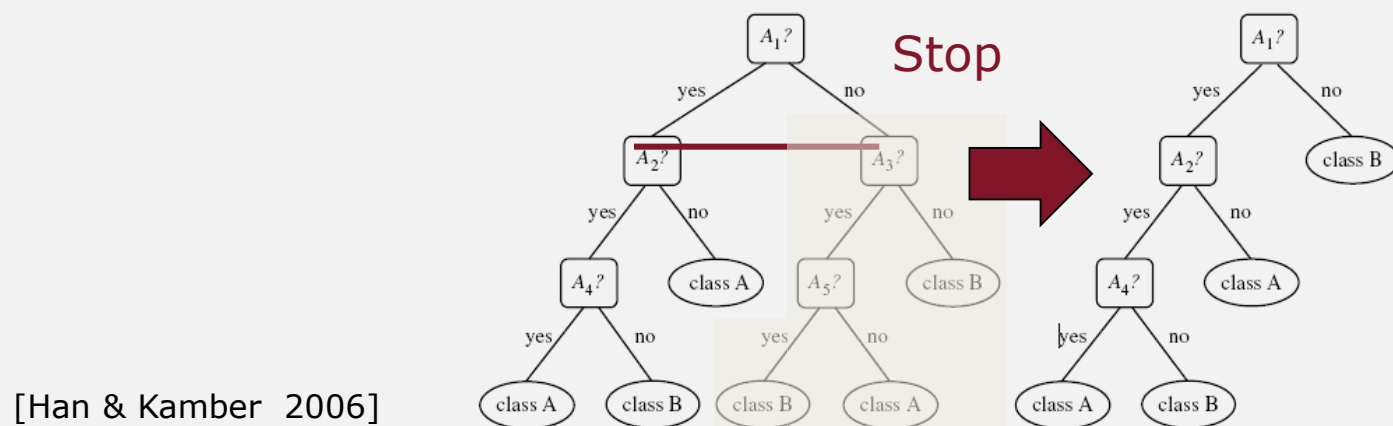


- Two strategies for “pruning” the decision tree:
 - Prepruning - stop growing a branch when information becomes unreliable
 - Postpruning - take a fully-grown decision tree and discard unreliable parts
- Postpruning preferred in practice - prepruning can “stop too early”

[Han & Kamber 2006]

Prepruning

- In prepruning approach, tree is “pruned” by **halting its construction early** (e.g. by deciding not to further split or partition subset of training tuples at given node).
- **Based on statistical significance test**: Stop growing tree when there is no statistically significant association between any attribute and class at particular node
- Leaf may hold **most frequent class** among subset tuples or **probability distribution** of those tuples.

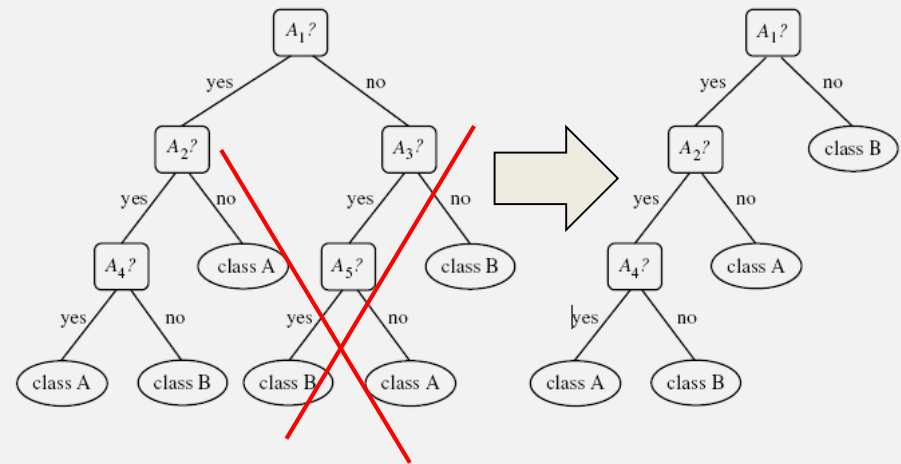


Post Pruning: Subtree Replacement & Raising

- In the **postpruning** approach, a tree is “pruned” **by removing subtrees** from a “fully grown” tree. Fully-grown tree shows all attribute interactions

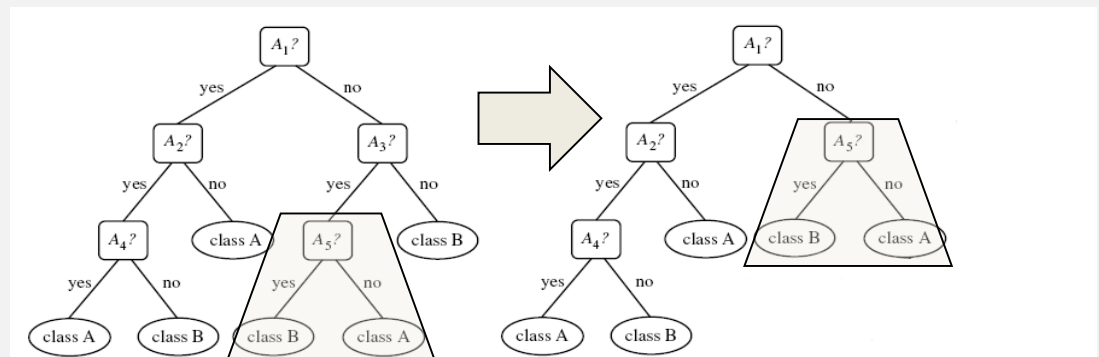
Two pruning operations:

1. Subtree replacement



2. Subtree raising

[Han & Kamber 2006]



1.6 Regression

- Variable and object space
- Concepts of regression
- Regression analysis techniques

Most common business analytics jobs

Problem	Business Perspective	Techniques
Find Clusters/Outliers	<ul style="list-style-type: none"> Are there different types of users Can we put different products together into distinct/different groups? 	<ul style="list-style-type: none"> Clustering Outlier-Analysis
Find Relationships	<ul style="list-style-type: none"> If a customer buys product A, what does he buy next? Which product sets belong together? 	<ul style="list-style-type: none"> Association Analysis
Predict Classes	<ul style="list-style-type: none"> Is this customer solvent or not? Will this customer send back this shipping or not? 	<ul style="list-style-type: none"> Decision Trees Logistic Regression Support-Vector Machines
Predict Values	<ul style="list-style-type: none"> Does a new label increase the? Is there a relationship between Sales and Commercials? 	<ul style="list-style-type: none"> Regression Support-Vector Machines
Predict Developments	<ul style="list-style-type: none"> How will the value of our products develop? What future developments are likely? 	<ul style="list-style-type: none"> Time-Series Forecasting



There will be further advanced techniques we will discuss in „Advanced Analytics with R“, another lecture covering topics like e.g. ANN, FP Growth techniques, Expectation Maxiization etc.

Regression Analysis

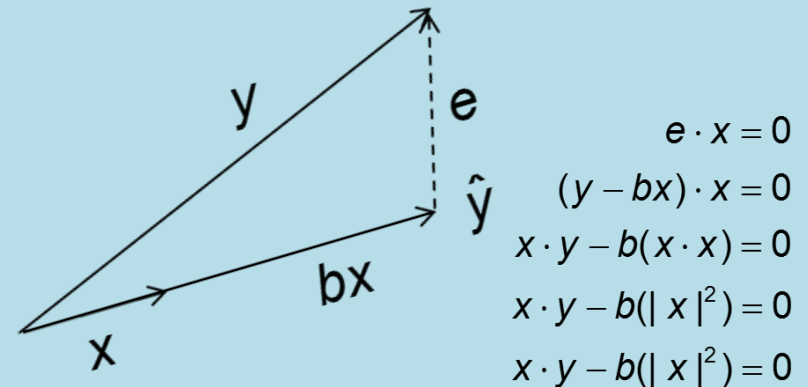
➤ Describes how specific phenomena relate to each other.

- Used to predict the value of one variable based on the value of other variables
- Independenten variables can be affected by each other but it does not mean that this dependency is both ways as is the case with correlatin analysis

Example:

$\text{Whisky_Consume} = X1 \cdot \text{Age} + X2 \cdot \text{Income} + e$

How much whisky someone drinks depends on age and income



Use Cases:

- **Consumer Analysis:** Levels of customer satisfaction affects customer loyalty
- **Pricing:** How neighbourhood and size affect the listing price of houses
- **Matchmaking:** Find the love of your life via online dating ;-)

$$Y = \text{deterministic component} + \text{stochastic component}$$

Prediction

- Univariate
- Multivariate

Deterministic Part

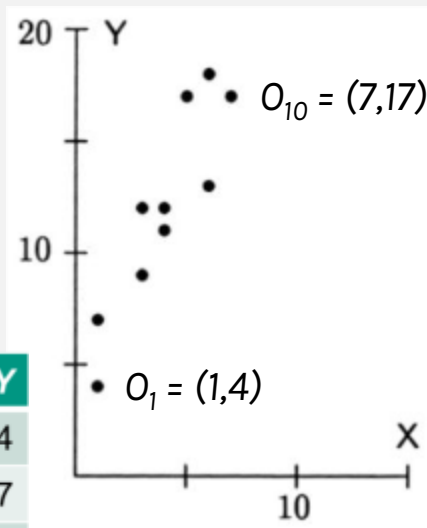
- Linear
- Nonlinear
- Smoothed

Stochastic Part

- Distribution
- Heterogeneity
- Auto-Correlation
- Nested data (Random Effects)
- Random Noise

Variable Space and Object Space

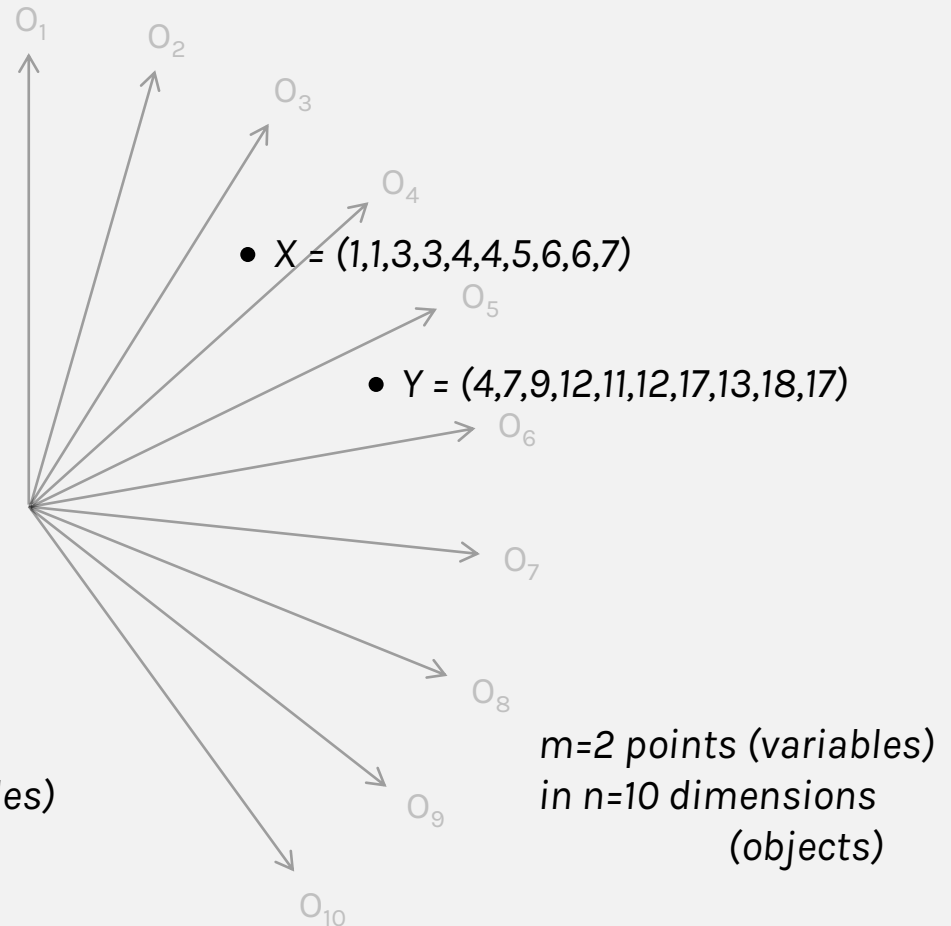
Variable Space



	X	Y
O ₁	1	4
O ₂	1	7
O ₃	3	9
O ₄	3	12
O ₅	4	11
O ₆	4	12
O ₇	5	17
O ₈	6	13
O ₉	6	18
O ₁₀	7	17

*n=10 points (objects)
in m=2 dimensions (variables)*

Object Space

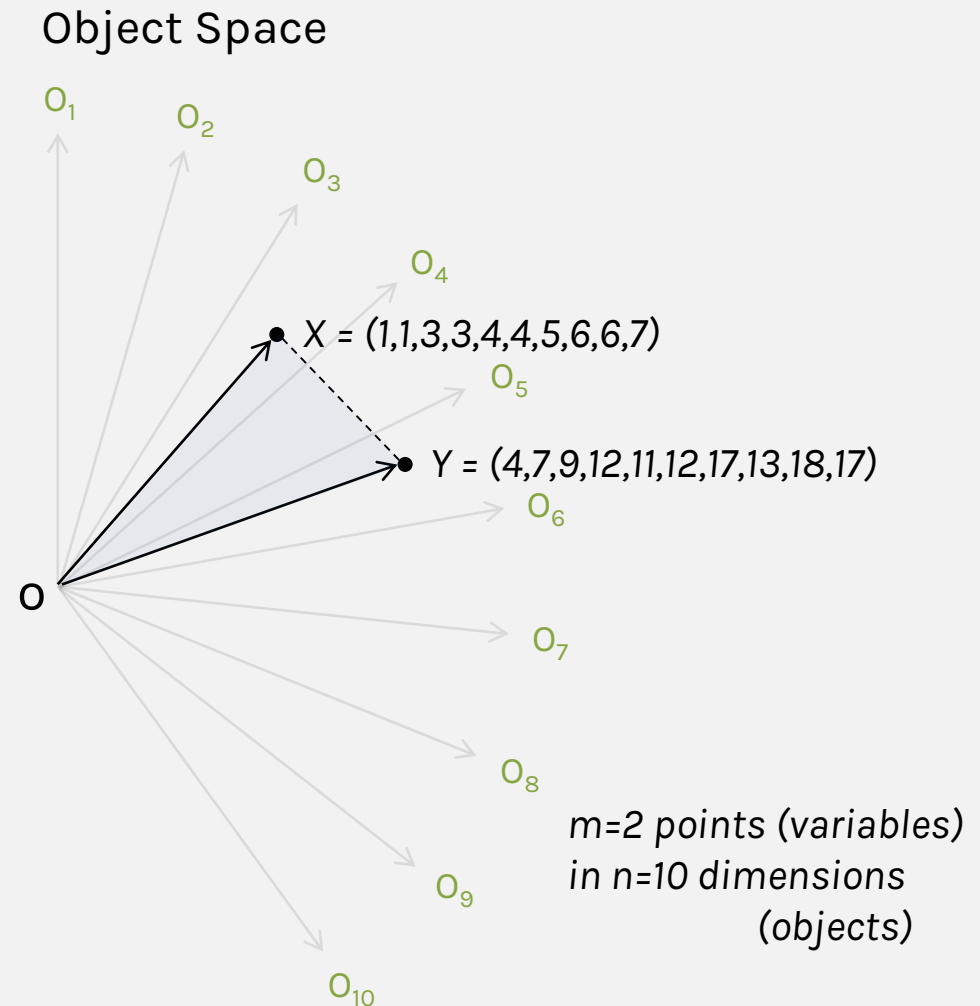


*m=2 points (variables)
in n=10 dimensions
(objects)*

Variable Space and Object Space

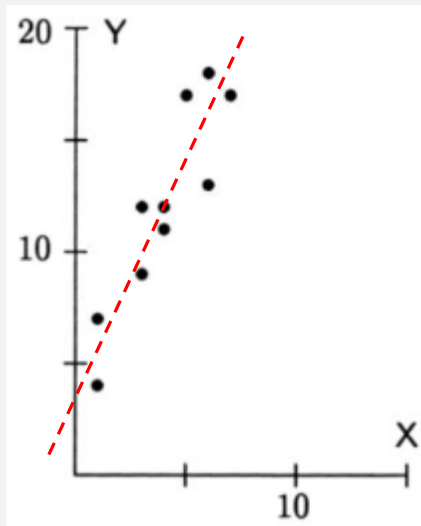
10-dimensional space is impossible to visualize.

However: If we have two vectors (variables), we can concentrate on the 2-dimensional space (plane) that they span (together with the origin O)



Bivariate Linear Regression I

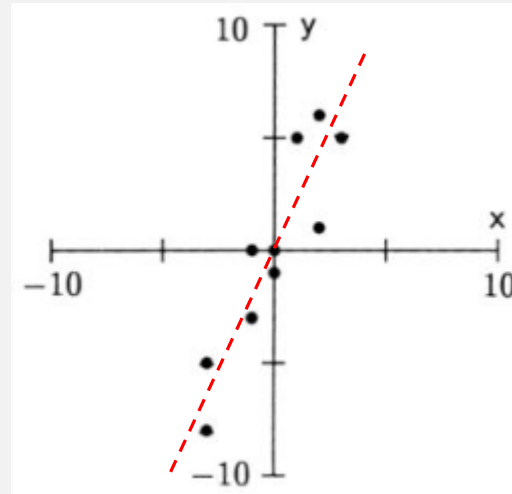
Variable Space



algebraic:

$$\hat{Y} = a + bX$$

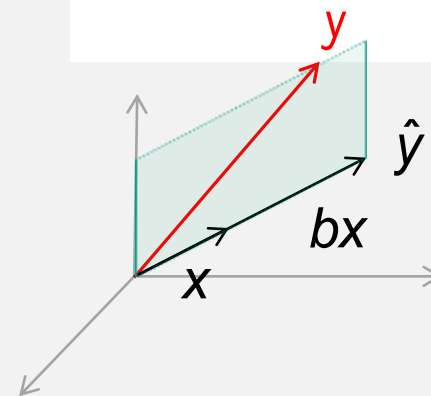
Centered: $x = X - \bar{X}, y = Y - \bar{Y}$



algebraic:

$$\hat{y} = bx$$

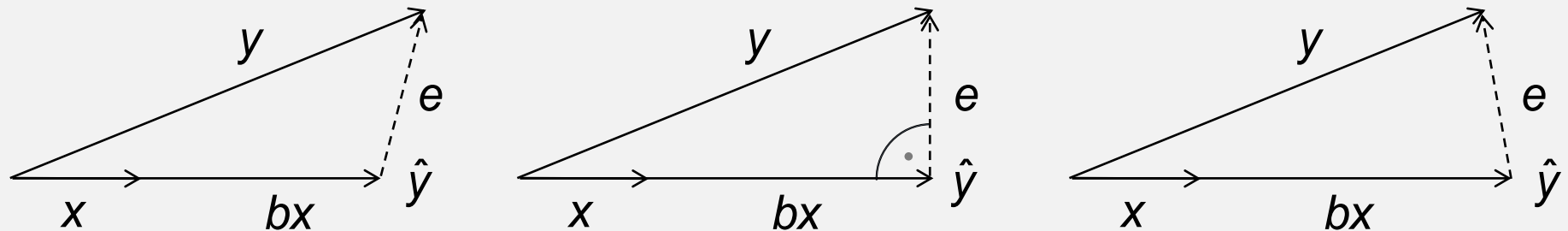
Object Space



vectorial:

$$\hat{y} = bx$$

Bivariate Linear Regression II



The best estimator \hat{y} is the one with the smallest error e . e is minimal if e is orthogonal to \hat{y} (and x):

$$e \perp x$$

Dot product must be zero:

$$e \cdot x = 0$$

$$(y - bx) \cdot x = 0$$

$$x \cdot y - b(x \cdot x) = 0$$

$$x \cdot y - b(|x|^2) = 0$$

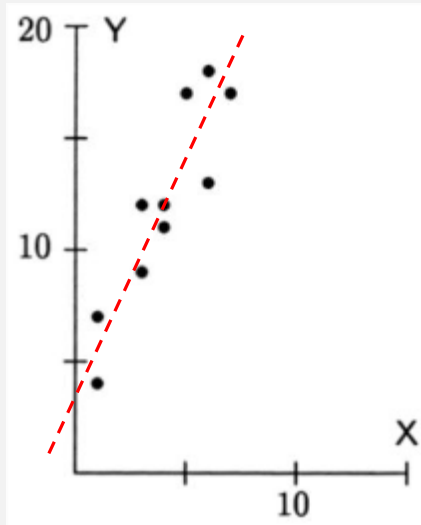
$$x \cdot y - b(|x|^2) = 0$$

$$b = \frac{x \cdot y}{|x|^2} = \frac{\sum_i x_i y_i}{\sum_i x_i^2}$$

This is the well-known regression formula obtained from the ordinary least squares approach (OLS) using differential calculus.

Bivariate Linear Regression III

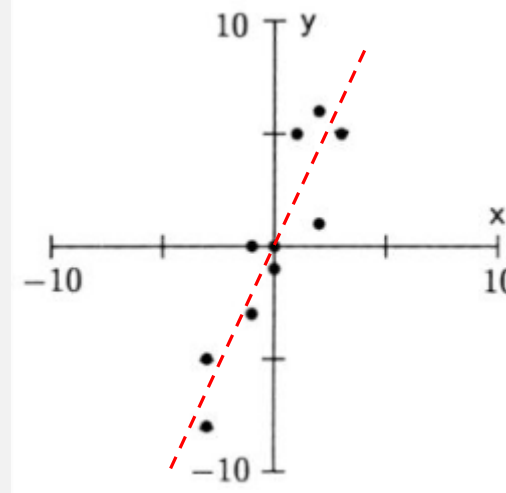
Variable Space



algebraic:

$$\hat{Y} = a + bX$$

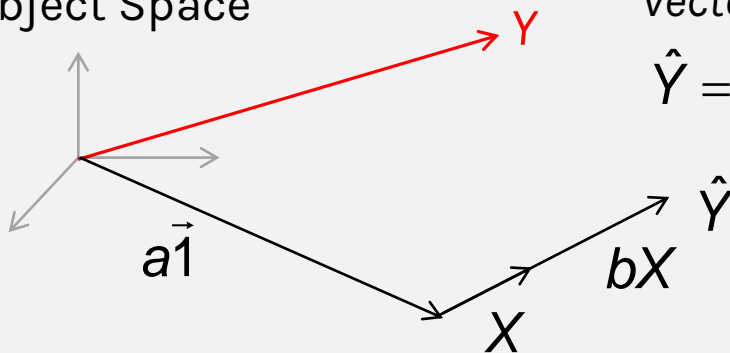
Centered: $x = X - \bar{X}, y = Y - \bar{Y}$



algebraic:

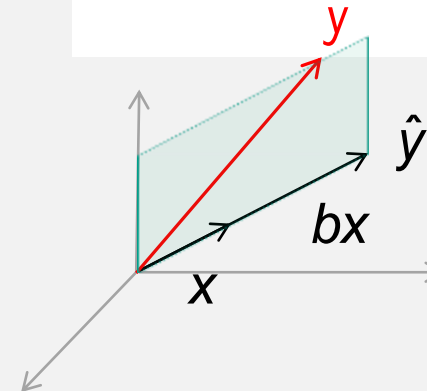
$$\hat{y} = bx$$

Object Space



vectorial:

$$\hat{Y} = a\vec{1} + b\vec{X}$$

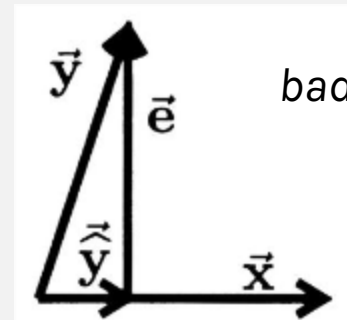
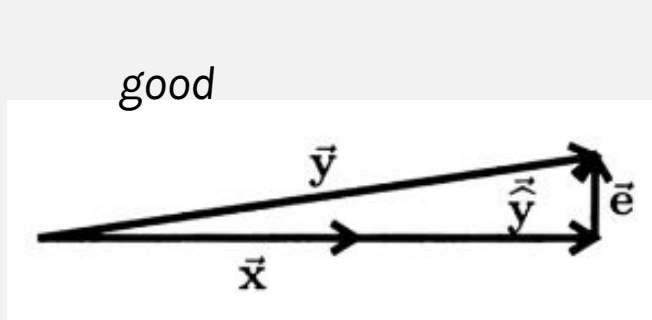


vectorial:

$$\hat{y} = b\vec{x}$$

Goodness of Fit I

- Regression predictor \hat{y} agrees with y as well as possible
- But this agreement can be good or poor
→ a measure is required to assess the quality of the fit



$$\cos \angle(\hat{Y}, Y) = \frac{|\hat{Y}|}{|Y|} := R$$

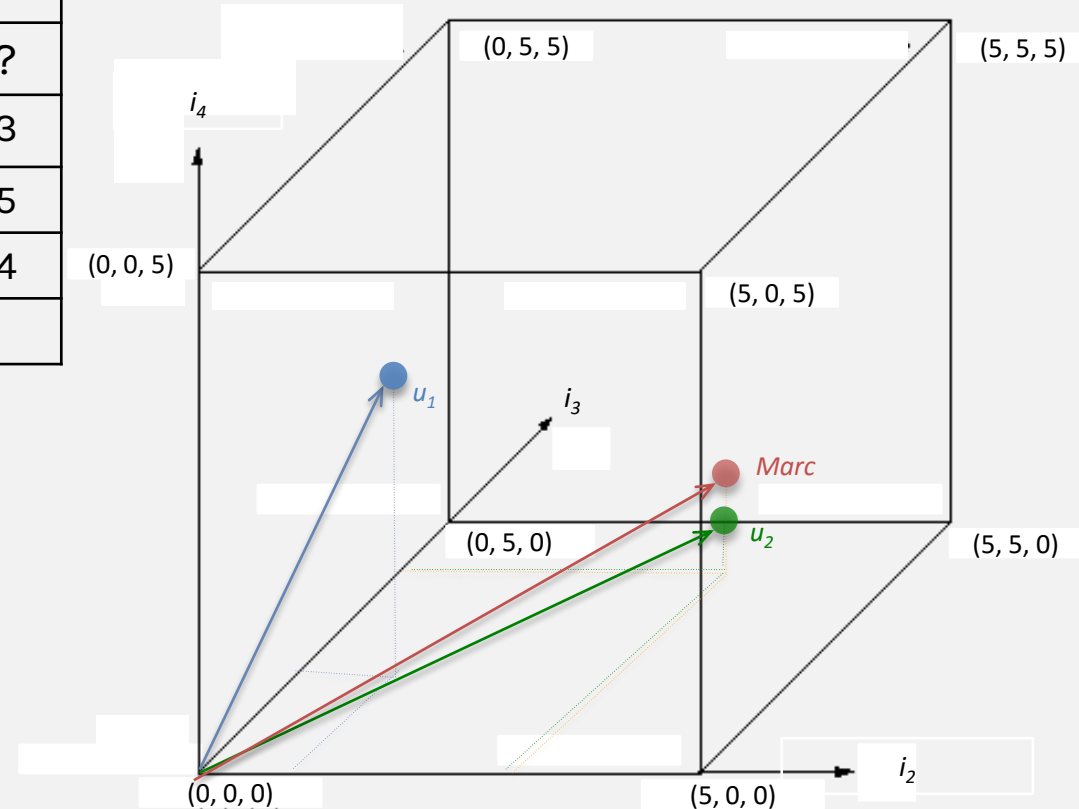
Correlation Coefficient
(Pearson's r)

- Either the angle between the vectors or their relative length can be used as a measure:

Similarity and correlation

	-- -	i_2	i_3	i_4	i_5
Marc		3	4	2	?
u_1		1	2	4	3
u_2		3	4	1	5
u_3		3	1	5	4
...					

- **Who is similar to Marc?**
- Marc's preferences are similar to the ones of u_2 and differ strongly from u_1 's
- → Marc's row vector (the point the vector ends) is much **closer** to the one of u_2 compared to the one of u_1
- **We measure closeness as correlation (or cosine-similarity)**



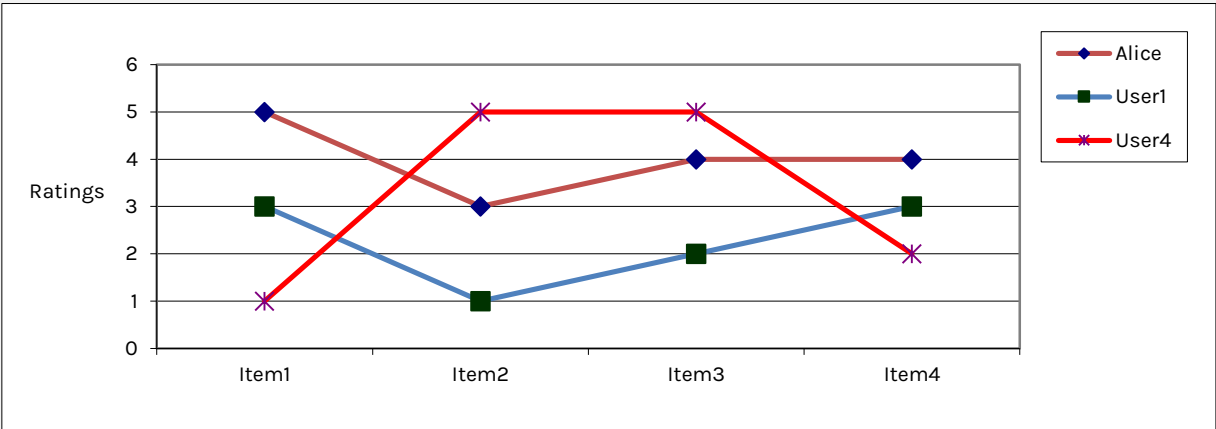
Measure user similarity to find the k nearest neighbors

$sim(u_o, u_u) = r(u_o, u_u)$, similarity values between -1 and 1

Computed on mutually ranked items

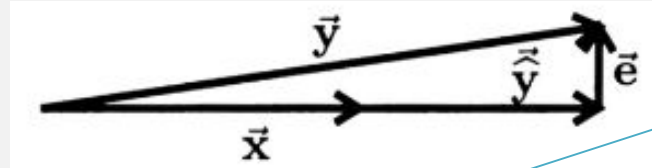
	Item1	Item2	Item3	Item4	Item5
Alice	5	3	4	4	?
User1	3	1	2	3	3
User2	4	3	4	3	5
User3	3	3	1	5	4
User4	1	5	5	2	1

$sim = 0,85$
 $sim = 0,70$
 $sim = -0,79$



Goodness of Fit II

$$R := \cos \angle(\hat{y}, y) = \frac{|\hat{y}|}{|y|}$$



Pythagoras:

$$|\hat{y}|^2 = |y|^2 - |e|^2$$

Recall:

If y is centered ($y_i = Y_i - \bar{Y}$)

$$|y| = \sqrt{y_1^2 + y_2^2 + \dots + y_n^2}$$

$$|y|^2 = y_1^2 + y_2^2 + \dots + y_n^2 \quad \text{„Sum of Squares“ } SS_y$$

$$s_y = \sqrt{\frac{1}{n-1}} |y| = \text{const} |y| \quad \text{Std. Dev.}$$

$$s_y^2 = \text{const} |y|^2 \quad \text{Variance}$$

$$SS_{\text{regression}} = SS_{\text{total}} - SS_{\text{residual}}$$

$$SS_{\text{regression}} = |\hat{y}|^2 = R^2 |y|^2 = R^2 SS_{\text{total}}$$

$R^2 = \frac{|\hat{y}|^2}{|y|^2}$ is called “coefficient of determination” and measures the „percentage of variance explained“

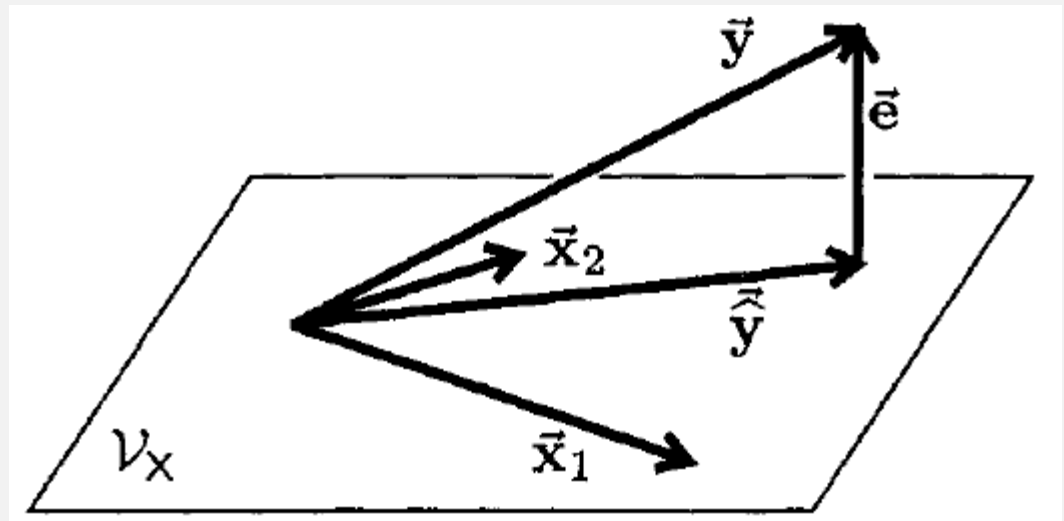
Multiple Regression I

- Output variable estimated by a linear combination of input variables

$$\hat{Y} = a\vec{1} + b_1X_1 + b_2X_2 + \dots + b_pX_p$$

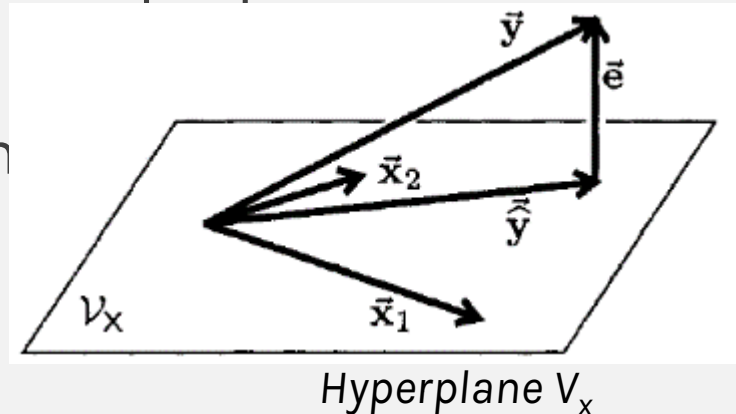
- Again, the core is fitting

$$\hat{y} = b_1x_1 + b_2x_2 + \dots + b_px_p$$



Multiple Regression II

- Now, we need to find the orthogonal projection of Y onto a hyperplane V_x
- Again, we need to find the minimum
- We know $e^{\wedge}x_1$ and $e^{\wedge}x_2$



$$x_1 \cdot e = 0,$$

$$x_1 \cdot (y - \hat{y}) = 0,$$

$$x_1 \cdot (y - b_1 x_1 - b_2 x_2) = 0,$$

$$x_1 \cdot y - b_1 x_1 \cdot x_1 - b_2 x_2 \cdot x_1 = 0$$

With the same for x_2 , we come to

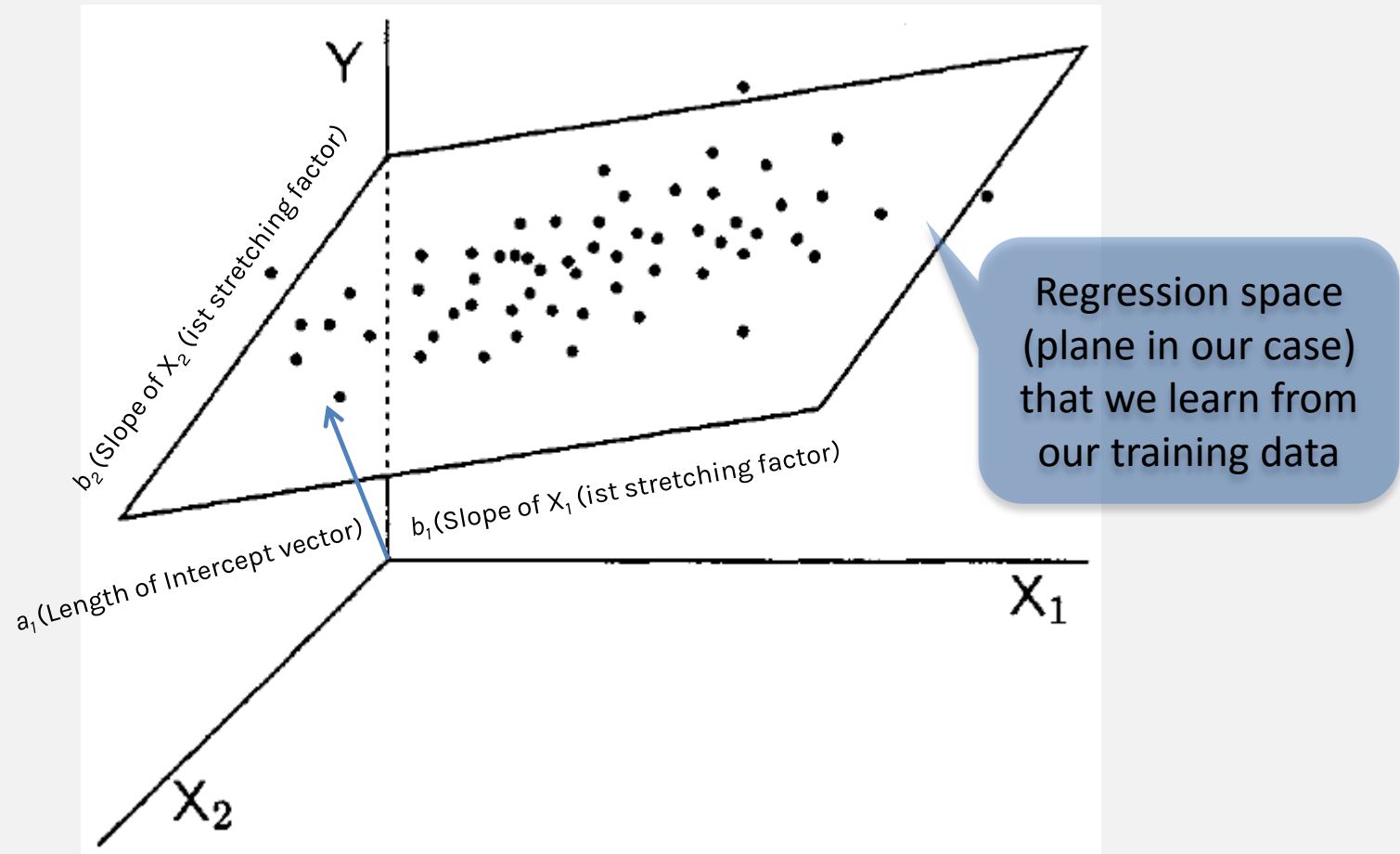
$$b_1(x_1 \cdot x_1) + b_2(x_1 \cdot x_2) = x_1 \cdot y$$

$$b_1(x_1 \cdot x_2) + b_2(x_2 \cdot x_2) = x_2 \cdot y$$

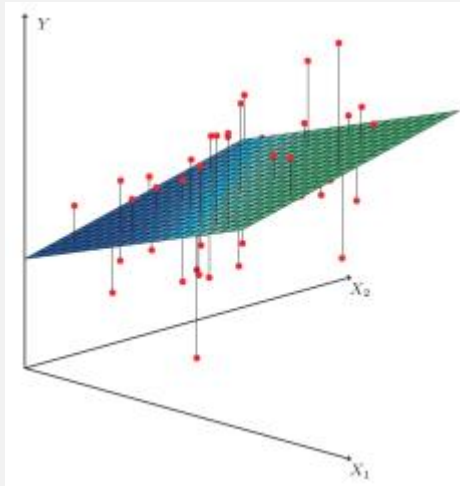
Known as the *normal equations*;
solving the system of equations
yields b_1, b_2

Multiple regression in variable space

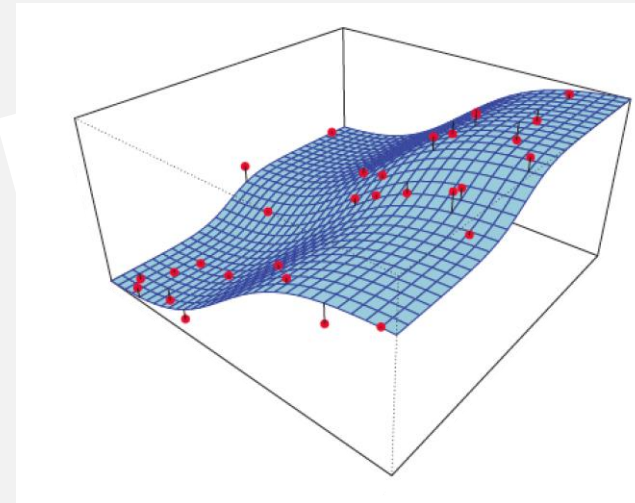
In multiple regression, we have a regression (hyper)plane and an **intercept** vector to learn, such that the sum of squared distances to the data points is minimized.



Non-linear regression (again) in variable space



- With two explanatory variables, we use a regression plane to predict Y



- Non-linearities by transforms (e.g., $X_1' := X_1^2 + 0.5X_1^3$)
- ... or learned (automatically): Kernel PC-Regression, Local Splines, Neural Networks, etc.

Summary

Take-away

- You know the most popular problems and techniques of modelling in analytics
- You also know how model performance can be measured and you understand the concepts behind the different approaches
- Finally, you can use (not implement) the most popular algorithms in R to analyse your data (e.g. make predictions, or find relationships), and understand the R output of these models

References

1. Hunt, E. B., Marin, J., & Stone, P. J. (1966) *Experiments in induction*. New York: Academic Press
2. Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., ... & Zhou, Z. H. (2008). *Top 10 algorithms in data mining*. *Knowledge and information systems*, 14(1), 1-37.