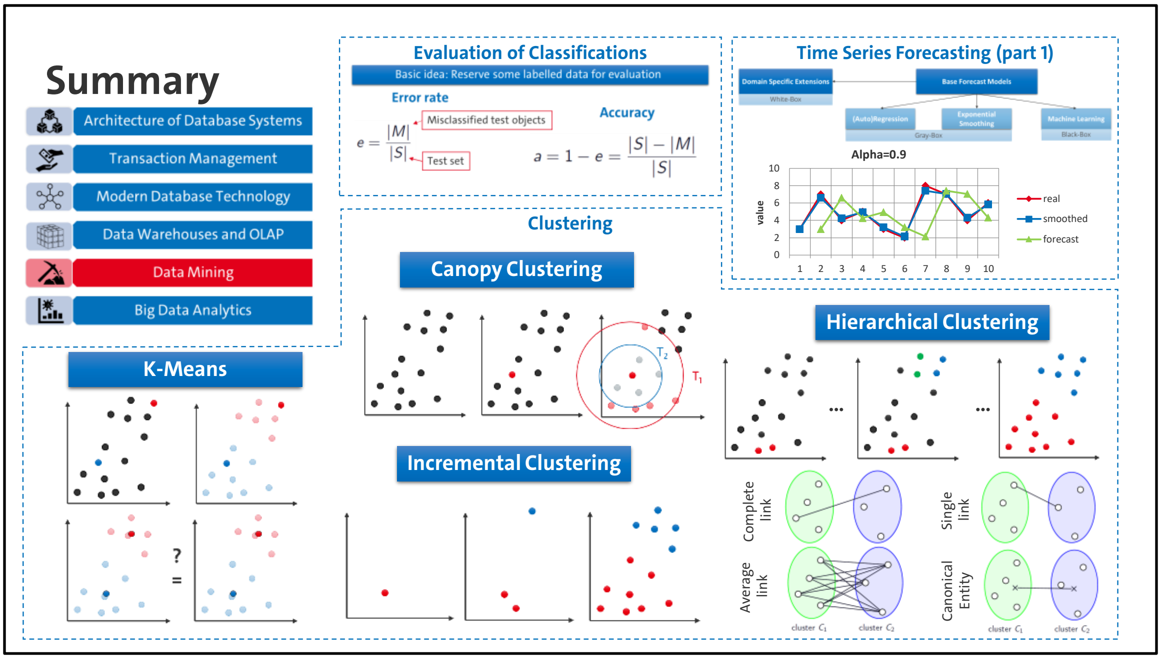

Summary
Architecture of Database Systems
Transaction Management
Modern Database Technology
Data Warehouses and OLAP
Data Mining
Big Data Analytics
Evaluation of Classifications
Basic idea: Reserve some labelled data for evaluation
Error rate
$e = \frac{|M|}{|S|}$
Misclassified test objects
Test set
Accuracy
$a = 1 - e = \frac{|S| - |M|}{|S|}$
Time Series Forecasting (part 1)
Domain Specific Extensions
Base Forecast Models
Alpha=0.9
value
real
smoothed
forecast
Clustering
Canopy Clustering
Incremental Clustering
K-Means
Hierarchical Clustering
Complete link
Single link
Average link
Canonical Entity

Model Types

Domain Specific Extensions ← **Base Forecast Models**

Base Forecast Models → **(Auto)Regression**, **Exponential Smoothing**, **Machine Learning**

Domain Specific Extensions
White-Box

HWT
(Energy Single Equation)
EGRV
(Energy, Multi Equation)
IFS
(Weather)

(Auto)Regression
Gray-Box

AR
MA
ARMA
ARIMA
SARIMA
ARMAX
MLR (Multiple Linear Regression)
VAR (Vector Autoregression)
MARS (Multivariate Adaptive Regression Splines)
GBM (Gradient Boosting Machine)

Exponential Smoothing
Gray-Box

SES
(Single Exponential Smoothing)
DES
(Double Exponential Smoothing)
TES/HoltWinters
(Triple Exponential Smoothing)

Machine Learning
Black-Box

BN
(Bayesian Networks)
SVR
(Support Vector Regression)
ANN
(Artificial Neural Networks)

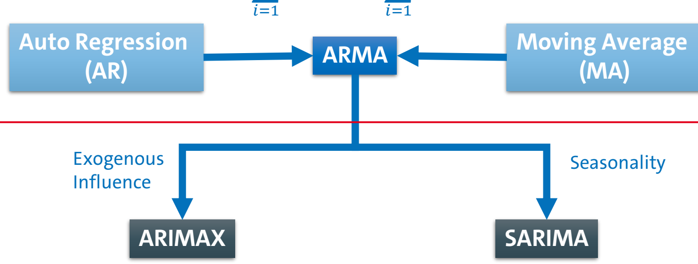
Data Mining

The ARMA Models

$$\hat{x}_t = c + \sum_{i=1}^{p} \phi_i \cdot x_{t-i}$$

$$\hat{x}_t = \sum_{i=1}^{p} \phi_i \cdot x_{t-i} + \sum_{i=1}^{q} \theta_i \cdot \varepsilon_{t-i} + c$$

$$\hat{x}_t = \sum_{i=1}^{q} \theta_i \cdot \varepsilon_{t-i}$$



$$\hat{x}_t = \sum_{i=1}^{p} \phi_i \cdot x_{t-i} + \sum_{i=1}^{q} \theta_i \cdot \varepsilon_{t-i} + c + \sum_{i=1}^{b} \gamma_i \cdot W_i$$

ARMA Models

t	1	2	3	4	5	6	7	8	9	10
x_t	3	7	4	5	3	2	8	7	4	6
AR	/	/	0.7*7+0.3*3 =**5.8**	0.7*4+0.3*7 =**4.9**	0.7*5+0.7*4 =**4.7**	3.6	2.3	6.2	7.3	4.9
MA	/	/	0.5*0+0.5*0 =**0**	0.5*0+0.5*4 =**2**	0.5*4+0.5*3 =**3.5**	1.25	0.125	4.3125	5.28125	0.703125
ε_i	0	0	4-0=**4**	5-2=**3**	3-3.5=**-0.5**	0.75	7.875	2.6875	-1.28125	5.296875

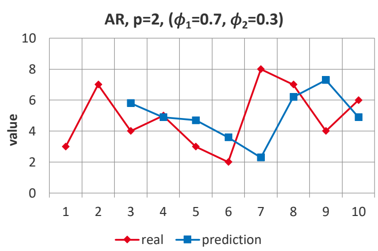
Box and Jenkins Methodology

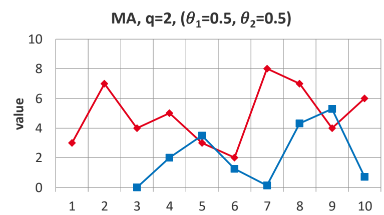
- Modeling time series with AutoRegression

Classification and Model Hierarchy

AR(p): $\hat{x}_t = c + \sum_{i=1}^{p} \phi_i \cdot x_{t-i}$
AutoRegression

MA(q): $\hat{x}_t = \sum_{i=1}^{q} \theta_i \cdot \varepsilon_{t-i}$
Moving Average
$\varepsilon_0, \cdots, \varepsilon_q = 0, \varepsilon_i = x_{t-i} - \hat{x}_{t-i}$

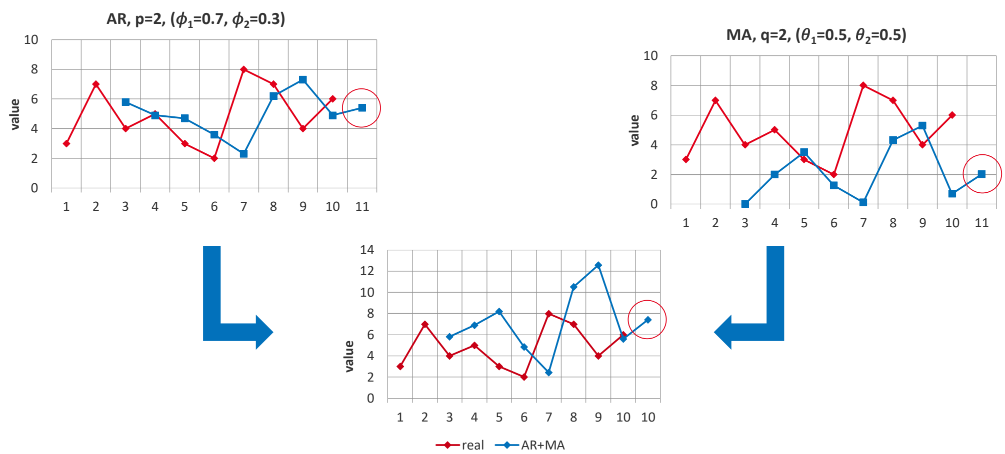




Data Mining

Here: c=0
Box and Jenkins do not assume trend and season components, but regard time series modelling as a stochastic process.

AR+MA



Exercise ARMA

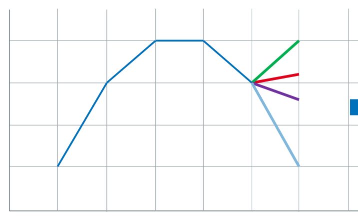
ϕ_1=0.7, ϕ_2=0.3
θ_1=0.5, θ_2=0.1

t	1	2	3	4	5	6
x_t	1	3	4	4	3	?
AR						
MA						
ε_i						

$$\hat{x}_t = \sum_{i=1}^{p} \phi_i \cdot x_{t-i}$$

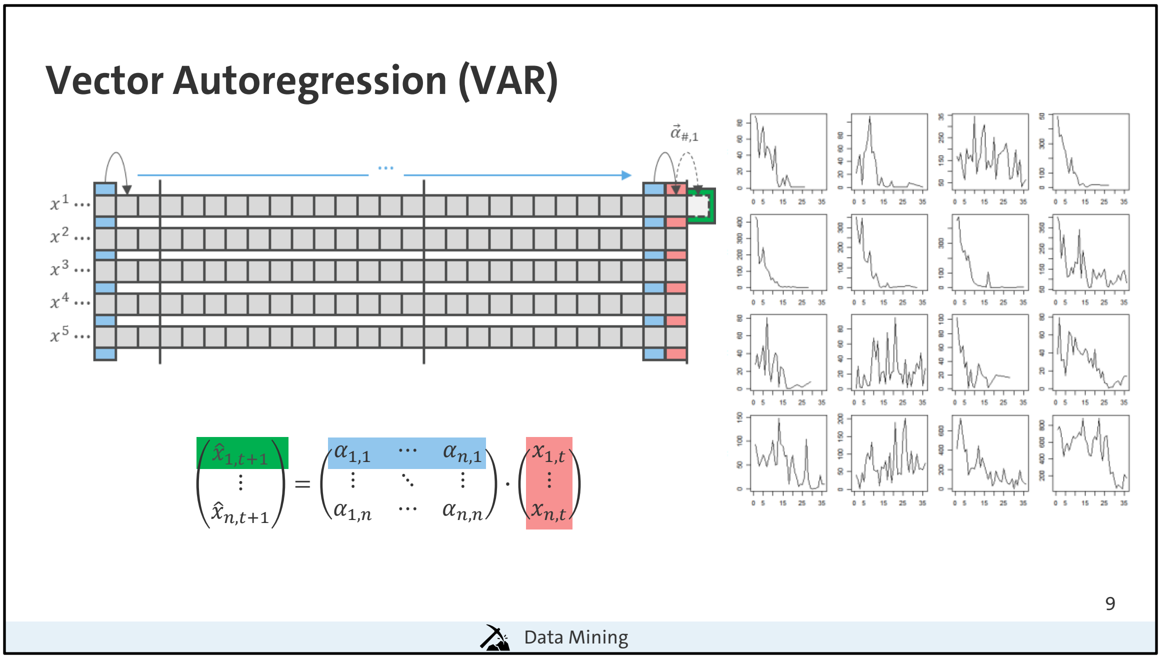
$$\hat{x}_t = \sum_{i=1}^{q} \theta_i \cdot \varepsilon_{t-i}$$

$$\varepsilon_0, \cdots, \varepsilon_q = 0, \varepsilon_i = x_{t-i} - \hat{x}_{t-i}$$



Which one is the ARMA prediction?

7



- Multivariate Modeling Technique
    - Based on the assumption, that time series from the same domain influence each other
    - Capable of compensating unexplainable behavior of individual time series
    - Can be extended by external influences
- Individual Model Optimization
    - In practice every line in the parameter matrix is implemented as an individual optimization process

Further Rreading

Riise, T., & Tjostheim, D. (1984). Theory and practice of multivariate arma forecasting. *Journal of Forecasting*, *3*(3).

Model Types

Domain Specific Extensions ← **Base Forecast Models**

White-Box

HWT
(Energy Single Equation)
EGRV
(Energy, Multi Equation)
IFS
(Weather)

Base Forecast Models → (Auto)Regression, Exponential Smoothing, Machine Learning

(Auto)Regression | **Exponential Smoothing** — Gray-Box

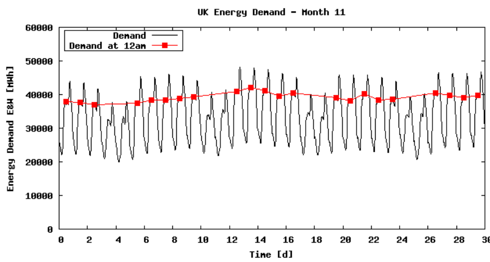
AR
MA
ARMA
ARIMA
SARIMA
ARMAX
MLR (Multiple Linear Regression)
VAR (Vector Autoregression)
MARS (Multivariate Adaptive Regression Splines)
GBM (Gradient Boosting Machine)

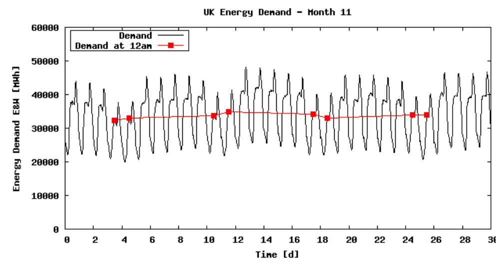
SES
(Single Exponential Smoothing)
DES
(Double Exponential Smoothing)
TES/HoltWinters
(Triple Exponential Smoothing)

Machine Learning — Black-Box

BN
(Bayesian Networks)
SVR
(Support Vector Regression)
ANN
(Artificial Neural Networks)

EGRV Model (Engle, Granger, Ramanathan, and Vahid-Arraghi)



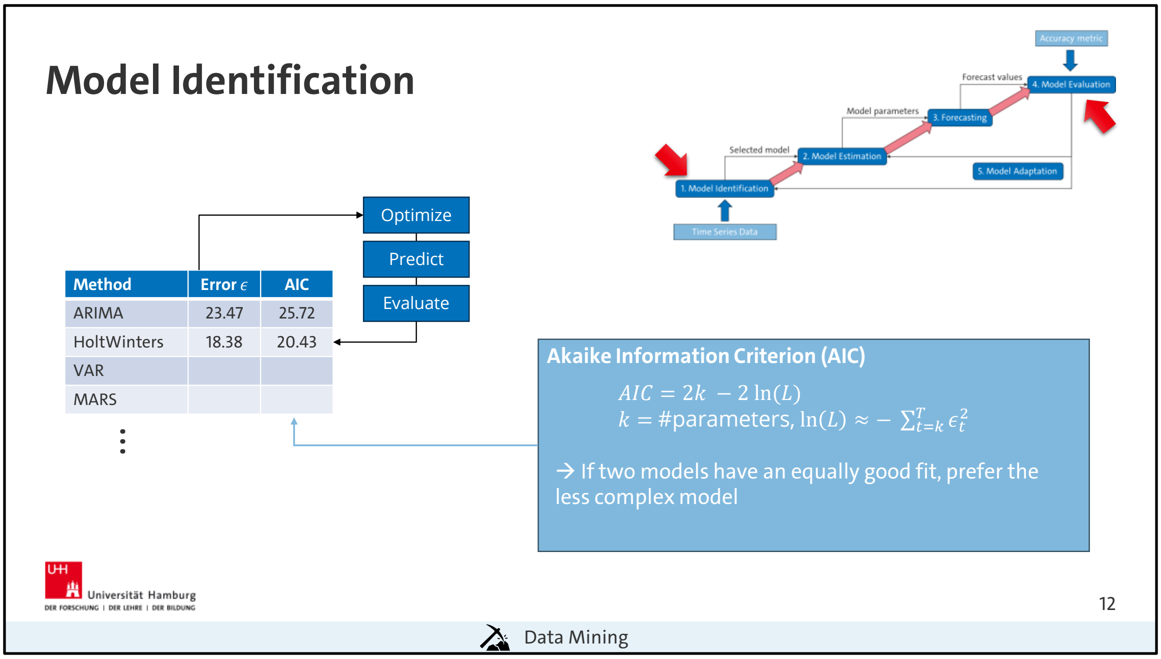


Ramanathan, R., Engle, R., Granger, C. W. J., Vahid-Araghi, F., & Brace, C. (1997). Short-run forecasts of electricity loads and peaks. *International Journal of Forecasting*, *13*(2).

Data Mining

Multi Equation Model

- One model for each hour (half hour, etc.) of a day (separation in weekdays/weekends)
- White-Box model tailor-made for energy demand
- Decomposition leads to almost constant time series that are easy to predict
- Basic idea works for all types of time series that follow certain pattern

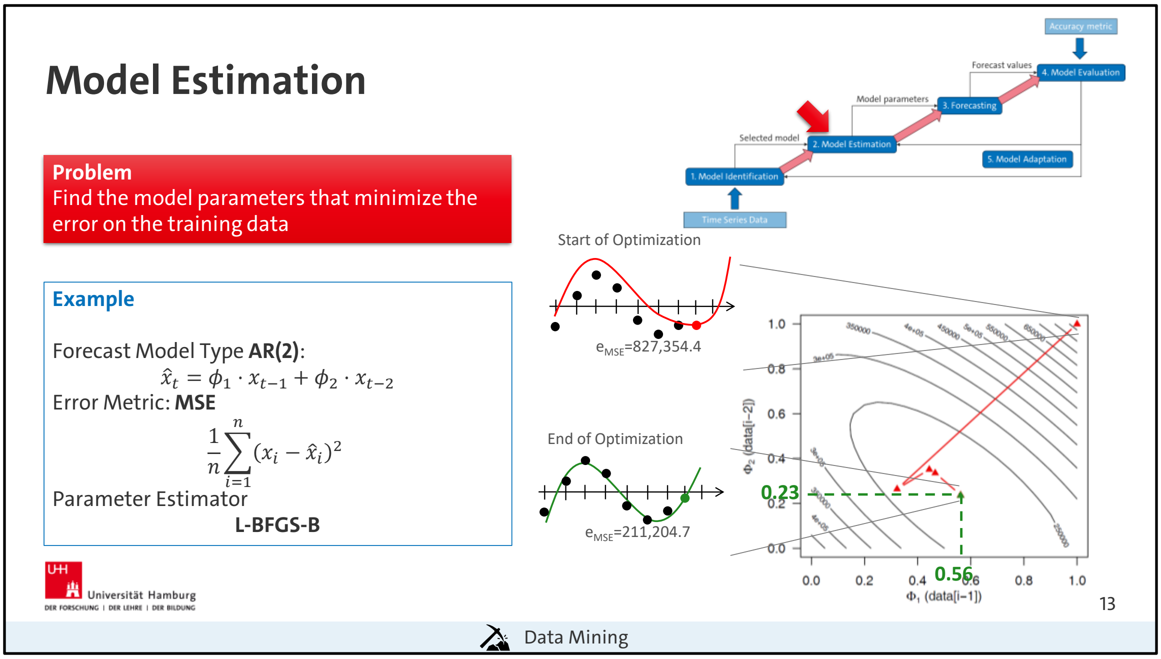


(Semi)Automatic Model Identification

- Test for every available model, how it performs on a test section of the current time series
- Use the error or more complex measures like AIC and BIC to find the optimal model

Akaike Information Criterion (AIC)

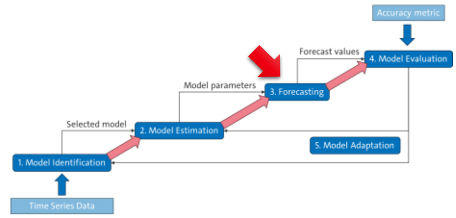
- A more sophisticated way to decide which is the optimal model
- Use the model with the lowest AIC
- Use the fit of the model to the training data and the model complexity
- L → Maximum likelihood



AR(2): Auto Regression with p=2
MSE: Mean Squared Error
L-BFGS-B: Limited memory Broyden–Fletcher–Goldfarb–Shanno with box constraints

Original BFGS was published independently by each of the 4 authors:
Broyden: https://doi.org/10.1093/imamat%2F6.1.76
Fletcher: https://doi.org/10.1093/comjnl%2F13.3.317
Goldfarb: https://doi.org/10.1090/S0025-5718-1970-0258249-6
Shanno: https://doi.org/10.1090/S0025-5718-1970-0274029-X

Model Usage



Model based Prediction
- Application of the identified and trained (optimized) model
- Typically very fast compared to the training process

Horizon
- Describes the number of forecasted values
- One step ahead (only one value), long range forecasting (many values, up to several seasons)

Example (Long Range)
- Single Exponential Smoothing, horizon 10 values
- Continue forecast calculation based on already calculated forecast values

Base:

$$x_1^* = x_1$$
$$x_t^* = \alpha \cdot x_t + (1-\alpha) \cdot x_{t-1}^*$$
$$\hat{x}_{t+1} = x_t^*$$

for(i in 1:10)

$$x_{t+i-1}^* = \alpha \cdot \hat{x}_{t+i-1} + (1-\alpha) \cdot x_{t-1}^*$$
$$\hat{x}_{t+i} = x_{t+i-1}^*$$

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Maintenance strategies

Error Threshold
- Define error threshold τ
- Update model parameter when $\text{Error} > \tau$
- May miss opportunities to improve the model performance or reestimate too often, based on whether τ is too high or too low

Time Threshold
- Reestimate the model parameter after a fixed number of update values
- May reestimate too often and without model improvement or tolerate too high errors within an interval

…as usual a combination works best

A somewhat higher error threshold
- And a somewhat longer time threshold

Time Series Storage

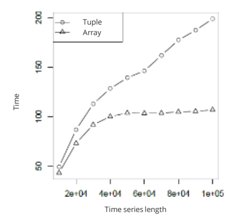
Relational Storage

ID	rec_date	sales
1	20.11.2021	3
1	21.11.2021	7
1	22.11.2021	4
1	23.11.2021	5
1	24.11.2021	3
1	25.11.2021	2
1	26.11.2021	8
1	27.11.2021	7
1	28.11.2021	4
1	29.11.2021	6
2	20.11.2021	7
2	21.11.2021	5
2	22.11.2021	8

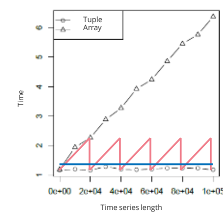
Array Storage

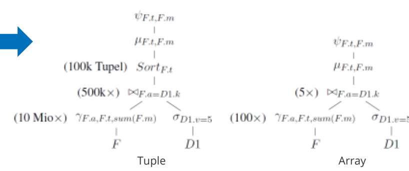
ID	start	interv	sales
1	20.11.2021	1 day	{3,7,4,5,3,2,8,7,4,6}
2	20.11.2021	1 day	{7,5,8,9,1,5,5,8,9,6}

Read performance
- Read entire data set into a forecast model
- Array storage outperforms tuple per value



Insert performance
- Append values to stored time series
- Tuple per value wins, less overhead
  - Counter steer by partitioning
- Ideal case, constant insert time





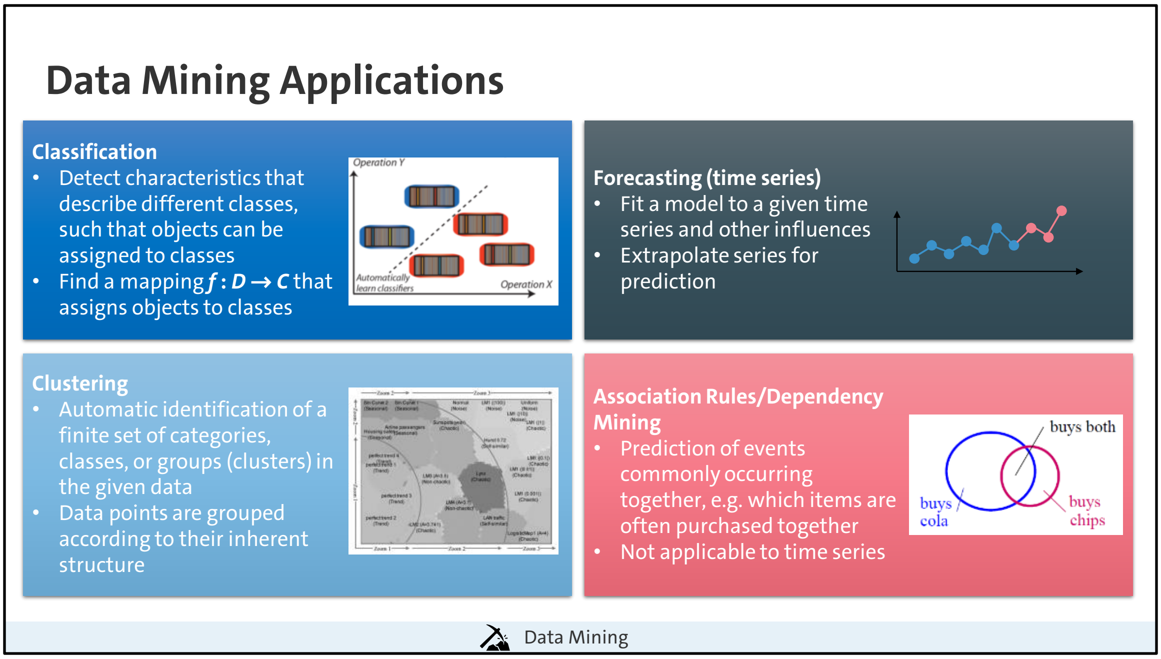
15

Relational Storage
- One tuple for each individual measure value
- Store alongside with
  - Time/date stamp
  - Time series identifier, e.g.
    - ID
    - Equipment
    - Sensor name

<u>Access</u>
- Usually no order guarantees
- Order by
  - Identifier first
  - Time last

Array Storage
- Store entire time series
  - Time series values are stored in an array
  - Alongside with
    - Time series identifier

        - Start
        - Time interval between values
- Constraints: Time series must be
    - Complete
    - Equidistant



Goal: Prediction of events commonly occurring together

Market basket analysis: Which items are often purchased together

- Placement of items in a store
- Layout of mail-order catalogues
- Targeted marketing campaigns

Association rules: Rules of the form a ∧ b ∧ . . . ∧ c → d ∧ e

Example: buys cola → buys chips

Challenge: Finding good combinations of premises and conclusions is a combinatorial problem

Association Rules

Example (transaction) database

trans-action	item
001	cola
001	chips
001	peanuts
002	beer
002	chips
002	cigarettes
...	...

I_k, i=2

trans-action	items
001	{chips, cola, peanuts}
002	{beer, chips, cigarettes}
003	{beer, chips, cigarettes, cola}
004	{beer, cigarettes}

T_k

D

I = SELECT DISTINCT item FROM example_table;

s({beer}) = 3/4

If (s_{cut} = ½) → s({beer}) is a frequent itemset

Definitions

Set of *n* different items *I* $\quad I = \{x_1, \ldots, x_n\}$

Itemset I_k $\quad I_k \subseteq I$

i-itemset $\quad I_k^i \subseteq I, \ |I_k^i| = i$

Transaction T_k $\quad T_k \subseteq I$

Database D $\quad D = \{(k, T_k) \mid k = 1, \ldots, m\}$

Support of an itemset = share of transactions which contain the itemset $\quad s(I_i) = \dfrac{|\{T_k \mid I_i \subseteq T_k\}|}{|D|}$

Frequent (strong, large) itemset $\quad s(I_i) \geq s_{cut}$

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Downward closure:

- Every subset of a frequent itemset is also a frequent itemset
- Every superset of a non-frequent itemset is also a non-frequent itemset

Association Rules

Association Rule
$X \rightarrow Y \; where \; X, Y \subseteq I, X \cap Y = \emptyset$

Support of a rule: Share of transactions which contain both, premise and conclusion of the rule

$$s(X \rightarrow Y) = s(X \cup Y) = \frac{|\{T_k | X \cup Y \subseteq T_k\}|}{|D|} = p(XY)$$

Confidence of a rule: Share of transactions supporting the rule from those supporting the premise

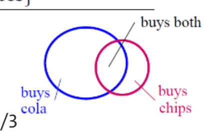
$$c(X \rightarrow Y) = \frac{s(X \cup Y)}{s(X)} = \frac{|\{T_k | X \cup Y \subseteq T_k\}|}{|\{T_k | X \subseteq T_k\}|} = p(Y|X)$$

Lift of a rule: Ratio of the observed support to that expected if *X* and *Y* were independent

$$lift(X \rightarrow Y) = \frac{s(X \cup Y)}{s(X) \times s(Y)} = \frac{p(Y|X)}{p(Y)}$$

trans-action	items
001	{chips, cola, peanuts}
002	{beer, chips, cigarettes}
003	{beer, chips, cigarettes, cola}
004	{beer, cigarettes}

Example:
[buys] chips → [buys] cola



s(chips→cola) = 2/4 = 0.5
c(chips→cola) = 0.5/(3/4) = 2/3
lift(chips→cola) = (2/3)/(0.75*0.5) = 1.78

Customers who buy chips are 1.78x as likely to also buy cola as compared to independent purchases.

Strong rule: High Support + High Confidence

Data Mining

- lift(X) > 1 → positive association
- lift(X) <1 → negative association
- lift(X) = 1 → items are independent

Detection of strong rules: Two pass algorithm

1. Find frequent (strong, large) itemsets (Apriori algorithm)
   - Necessary to generate rules with strong support
   - Uses the downward closure
   - Itemsets are ordered
2. Use the frequent itemsets to generate association rules
   - Find strong correlations in a frequent itemset

Apriori

I_k^1	#	$s(I_k^1)$
{chips}	3	0.75
{cola}	2	0.5
{peanuts}	1	0.25
{beer}	3	0.75
{cigarettes}	3	0.75

Step 1: Start with all itemsets of size one: I^1
Step 2: Select all items with sufficient support, e.g where $s(I_k)\geq 0.5$

I_k^2	#	$s(I_k^2)$
{chips, cola}	2	0.5
{beer, chips}	2	0.5
{chips, cigarettes}	2	0.5
{beer, cola}	1	0.25
{cigarettes, cola}	1	0.25
{beer, cigarettes}	3	0.75

Step 3: From the selected itemsets I^{n-1} generate larger itemsets I^n
→ *Already blocks some of the non-frequent itemsets, but not all of them*

I_k^2	#	$s(I_k^2)$
{chips, cola}	2	0.5
{beer, chips}	2	0.5
{chips, cigarettes}	2	0.5
{beer, cola}	1	0.25
{cigarettes, cola}	1	0.25
{beer, cigarettes}	3	0.75

Step 4: Remove itemsets which still contain a non-frequent subset
→ *Cannot have enough support because of downward closure*

I_k^3	#	$s(I_k^3)$
{beer, chips, cigar.}	2	0.5
{chips, cigar., cola}	1	0.25

Prune because of non-frequent subset

Step 5: *Repeat from step 2 until no further frequent itemsets can be generated*

D

trans-action	items
001	{chips, cola, peanuts}
002	{beer, chips, cigarettes}
003	{beer, chips, cigarettes, cola}
004	{beer, cigarettes}

Resulting frequent itemsets

i= 3: {beer, chips, cigarettes}
i=2: {chips, cola}, {beer, chips}, {chips, cigarettes}, {beer, cigarettes}
i=1: {beer}, {chips}, {cigarettes}, {cola}

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

19

Data Mining

Apriori: Finding frequent itemsets of increasing size (itemsets are ordered!)

- Step 4 in the example: no items to prune
- Note: Itemset {*beer, chips, cola*} is not constructed because we only combine sets of I^2_k that have the first element in common
- However, this set cannot be frequent because otherwise the set {*beer, cola*} would be in I^2_k (and then we would have constructed {*beer, chips, cola*})

Exercise Apriori

k	T_k
1	{butter, bread}
2	{butter, bread, cheese, wine}
3	{bread, soda}
4	{cheese, pencils, pasta}
5	{cheese, pasta, wine}

$s_{cut} = 0.4$

Generation of Strong Association Rules

- For all frequent itemsets I_j with $|I_j| \geq 2$ determine all non-empty subsets I_k for which

$$c = \frac{s(I_j)}{s(I_k)} \geq c_{min}$$

- Add rule $I_k \rightarrow Y$ with $Y = I_j - I_k$ to the rule set

Example

$s(\{chips\}) = 0.75$, $s(\{cola\}) = 0.5$, $s(\{chips, cola\}) = 0.5$

rule	confidence
$\{\text{cola}\} \rightarrow \{\text{chips}\}$	1.00
$\{\text{chips}\} \rightarrow \{\text{cola}\}$	0.67

Confidence

$$c(X \rightarrow Y) = \frac{s(X \cup Y)}{s(X)} = \frac{|\{T_k | X \cup Y \subseteq T_k\}|}{|\{T_k | X \subseteq T_k\}|} = p(Y|X)$$

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

22

Data Mining

Interesting association rules: Only those for which the confidence is greater than the support of the conclusion

$$c(X \rightarrow Y) > s(Y) \qquad (\equiv lift(X \rightarrow Y) > 1)$$

Rule modification: Confidence can be increased by shifting items from the conclusion to the premise

Negative border: Used to derive negative association rules (e.g. customers who buy cola and chips do not buy beer)

$$\{I_j \mid s(I_j) < s_{cut} \wedge \forall I_k \subset I_j \colon s(I_k) \geq s_{cut}\}$$

Apriori: Number of potential itemsets is exponential in the number of items

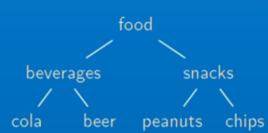
But:

- Data is sparse: $|T_i| << |I|$
- Itemsets are generated in separate scans of the database
- Size of generated itemsets grows monotonically
- Large itemsets are usually useless
- Only k scans required (k << |I|)

Modifications and Extensions of Apriori

Hierarchical Apriori

In addition to the base level of items, determine also frequent itemsets on a higher level in an is-a hierarchy
→ Sometimes regularities can only be found at higher levels of abstraction



Partitioned Apriori

1st step:
- Partition the database
- Compute locally frequent itemsets on each partition

2nd step:
- Determine the global support of all locally frequent itemsets
- Remove all itemsets which do not satisfy the minimal support

Heuristics:
- Idea: if an itemset is globally frequent it will be so locally in at least one partition

→ Second step deals with a superset of possible frequent itemsets

More

- Sampled transactions
- Incremental rule mining
- Non uniform support thresholds
- Class association rules
- Rule mining on relational data

Association Rules with Relational Data

- Relational data has to be transformed into transaction data
- Apriori requires categorical data → binning must be performed
- The same category can appear as value of different attributes
  - → Values must be combined with their attribute
  - → Attribute-value pairs are taken as items