Lecture 11:

Time Series Analysis

Outline

- Motivation
- Time series pre-processing
- Time series forecasting
- Pattern discovery
- Anomaly detection

Motivation

Problem: Let us suppose that we have weekly stock data for Dow Jones Index and we are interested to determine which stock will produce the greatest rate of return in the following week.

Data set: Dow Jones Index (UCI Machine Learning, provided by (Brown, Pelosi & Dirska, 2013) - 750 records, 16 attributes

Examples of DowJones Index stocks:

3M	MMM	Cisco Systems	CSCO
American Express	AXP	Coca-Cola	KO
Alcoa	AA	DuPont	DD
AT&T	T	ExxonMobil	XOM
Bank of America	BAC	General Electric	GE
Boeing	BA	Hewlett-Packard	HPQ
Caterpillar	CAT	The Home Depot	HD
Chevron	CVX	Intel	INTC
			IBM

Motivation

Example [Dow Jones Index   from   http://archive.ics.uci.edu/ml/datasets.html]
16 attributes
quarter: the yearly quarter (1 = Jan-Mar; 2 = Apr-Jun).
stock: the stock symbol (e.g. list on the previous slide)
date: the last business day of the work (this is typically a Friday)
open: the price of the stock at the beginning of the week
high: the highest price of the stock during the week
low: the lowest price of the stock during the week
close: the price of the stock at the end of the week
volume: the number of shares of stock that traded hands in the week
percent_change_price: the percentage change in price throughout the week
percent_change_volume_over_last_wek: the percentage change in the number of shares of stock that traded hands for this week compared to the previous week
previous_weeks_volume: the number of shares of stock that traded hands in the previous week

Motivation

Example [Dow Jones Index from http://archive.ics.uci.edu/ml/datasets.html]
16 attributes

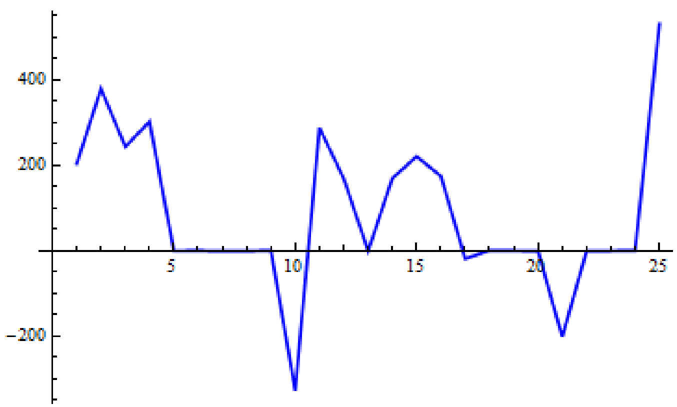
next_weeks_open: the opening price of the stock in the following week
next_weeks_close: the closing price of the stock in the following week
percent_change_next_weeks_price: the percentage change in price of the stock in the following week
days_to_next_dividend: the number of days until the next dividend
percent_return_next_dividend: the percentage of return on the next dividend

Problem: which stock will produce the greatest rate of return in the following week?

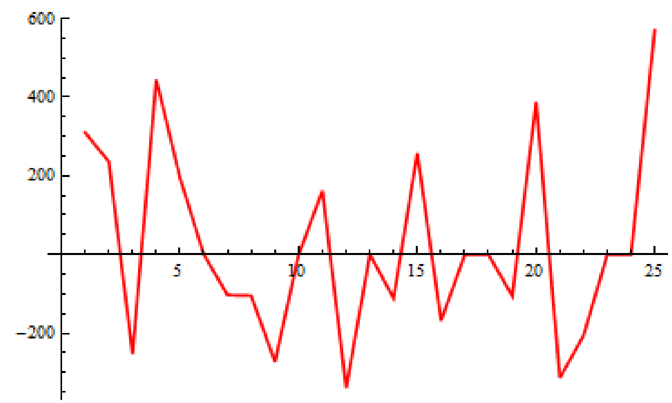
Motivation

Problem: which stock will produce the greatest rate of return in the following week?

Example [Dow Jones Index   from   http://archive.ics.uci.edu/ml/datasets.html]
16 attributes
percent_change_next_weeks_price: the percentage change in price of the stock in the following week



IBM

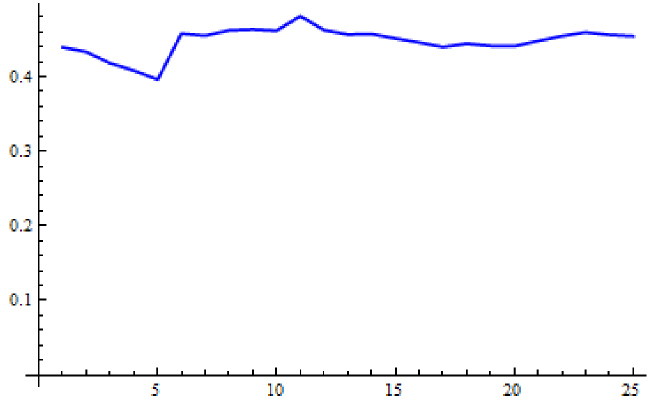


HP

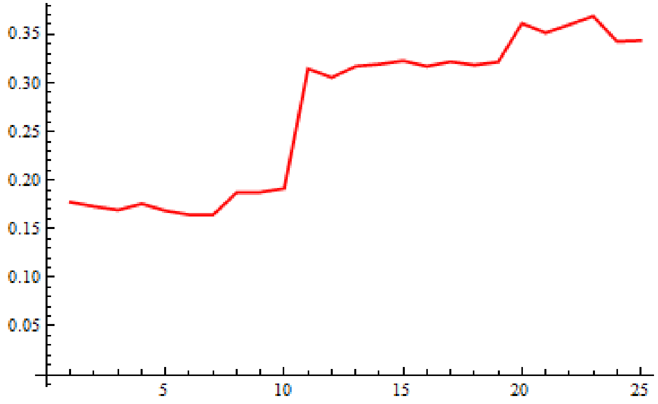
Motivation

Problem: which stock will produce the greatest rate of return in the following week?

Example [Dow Jones Index from http://archive.ics.uci.edu/ml/datasets.html]
16 attributes
percent_return_next_dividend: the percentage of return on the next dividend



IBM



HP

Motivation

Besides financial data there are other sources of time series:

- Sensor data:
  - Environmental data collected from various sensors (temperature, pressure, humidity
- Medical data
  - Electrocardiogram (ECG)
  - Electroencephalogram (EEG)
  - Real time monitoring data collected from patients in intensive care units
- Web log data (clickstream data)
  - Sequence of web page visits

Motivation

Besides financial data there are other sources of time series:

- Sensor data:
  - Environmental data collected from various sensors (temperature, pressure, humidity

    Task: predict values corresponding to future moments
- Medical data
  - Electrocardiogram (ECG)
  - Electroencephalogram (EEG)
  - Real time monitoring data collected from patients in intensive care units

    Task: identify anomalous behaviour
- Web log data (clickstream data)
  - Sequence of web page visits

    Task: identify usage patterns, user profiles

Time series

Example 1 (the percentage of return on the next dividend for first 10 weeks included in Dow Jones Index dataset)

0.177, 0.172 ,0.169, 0.175, 0.168, 0.164, 0.164, 0.187, 0.187, 0.191

The time moment is not explicitly specified. However it influences the values in the series, thus these values should be interpreted in the context of the time moments

- The time is a contextual attribute
- The percentage of the return is a behavioral attribute

Example 2 (temperature at noon during the last seven days)

21, 24, 23, 25, 22, 19, 20

The contextual attribute is the time, the behavioral attribute is the temperature

Time series

There are different types of time series

With respect to the time domain:

- Continuous (e.g. EEG)
- Discrete (they are called time sequences)

With respect to the number of behavioral attributes

- Univariate (one attribute)
- Multivariate (several attributes)

Time series pre-processing

Missing values

Problem:

- values corresponding to some time moments may be missing (e.g. faults of sensors)
- Especially when there are several behavioral attributes (collected by independent sensors) it is necessary to ensure the synchrony between data by filling in the missing values

Solution:

- The missing value can be estimated by interpolation
- Simplest case: linear interpolation

Time series pre-processing

Imputation of missing values by linear interpolation

Let $(y_1,y_2,\ldots,y_n)$ be the series of values of the behavioral attribute corresponding to the time moments $(t_1,t_2,\ldots,t_n)$

Let us suppose that there is a missing value at time moment t which is between t_i and t_{i+1}. By supposing that the behavioral attribute y varies linearly with t on $[t_{i\,,}\ t_{i+1}]$ we can estimate the value of y

$$y = y_i + \frac{t - t_i}{t_{i+1} - t_i}(y_{i+1} - y_i)$$

Time series pre-processing

Noise removal

Problem: the devices used to collect data (e.g. sensors) may be noise-prone, thus the time series may contain some artifacts generated during the collection process and which are not reflecting the true behavior in time of the recorded attribute

Approaches in dealing with the noise

- Binning
- Moving-Average Smoothing

Time series pre-processing

Binning (piecewise aggregate approximation)

Idea:

- the total time interval $[t_1, t_n]$ corresponding to a series $(y_1, y_2, \ldots, y_n)$ is divided in m subintervals containing each one k elements (m=n/k)
- each subinterval will be associated to one value computed as average of the values in the time series corresponding to the time moments included in the sub-interval

Remarks:

- it is supposed that the time moments of the initial series are equally spaced
- it reduces the number of available data by k (it is a type of lossy compression)

$$(t_1, t_2, \ldots, t_n) \rightarrow ((t_1, \ldots, t_k), (t_{k+1}, \ldots, t_{2k}), \ldots, (t_{(m-1)k+1}, \ldots, t_{mk})$$

$$(y_1, y_2, \ldots, y_n) \rightarrow (z_1, z_2, \ldots, z_m)$$

$$z_i = \frac{1}{k} \sum_{j=1}^{k} y_{(i-1)k+j}, \; i = \overline{1, m}$$

Time series pre-processing

Moving average smoothing

Idea: it reduces the loss of binning by using overlapping bins on which the averages are computed, i.e. the average is computed over the elements belonging to a window (which is moving over the time series)

$$(t_1, t_2, ..., t_n) \rightarrow ((t_1, ..., t_k), (t_2, ..., t_{k+1}), ..., (t_{(m-1)k+1}, ..., t_{mk})$$

$$(y_1, y_2, ..., y_n) \rightarrow (z_1, z_2, ..., z_m)$$

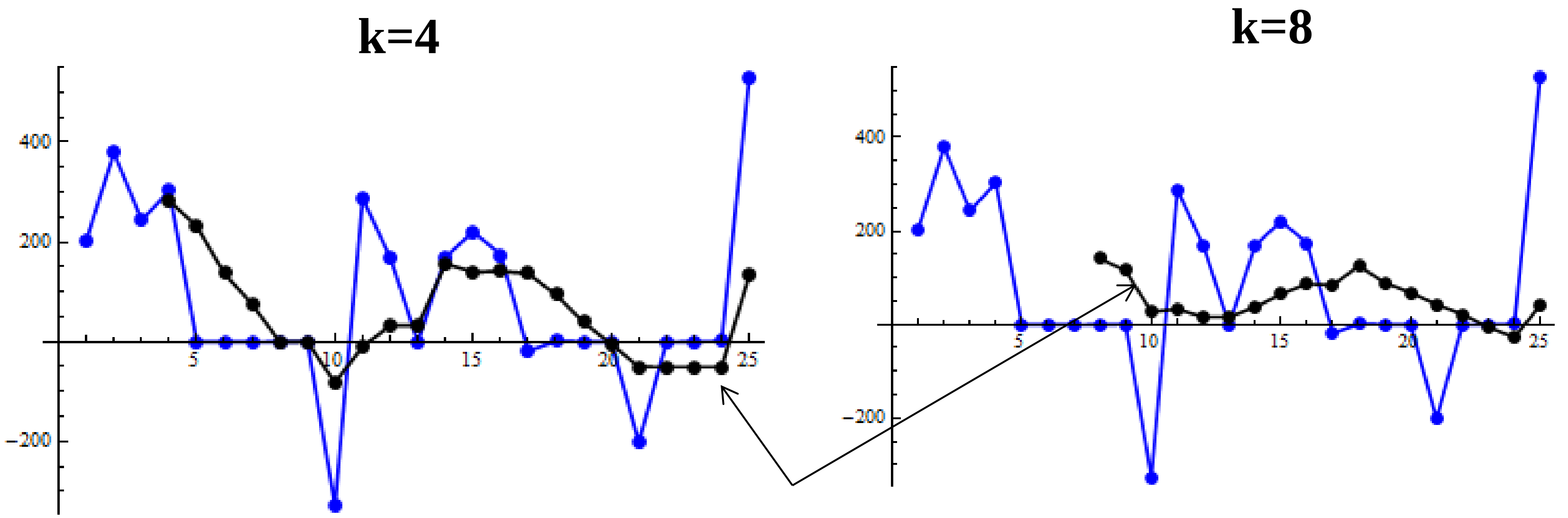
$$z_i = \frac{1}{k} \sum_{j=1}^{(m-1)k+1} y_{i+j-1}, \; i = \overline{1, m}$$

Remarks:

- The number of elements is reduced from *n* to *n-k+1*
- In a real time context, the average moving introduces a lag of k time moments
- Short term trends can be lost because of smoothing

Time series pre-processing

Example (Moving average smoothing)



Smoothed series (black)

Remarks:

- The number of elements is reduced from *n* to *n-k+1*
- In a real time context, the average moving introduces a lag of k time moments
- Short term trends can be lost because of smoothing

Time series pre-processing

Exponential smoothing

Idea: the smoothed value is defined as a linear combination of the current value and the previously smoothed value

$$z_i = \alpha \cdot y_i + (1-\alpha) \cdot z_{i-1}, \quad i = \overline{1,m}$$

$$z_0 = y_1$$

$$z_i = (1-\alpha)^i z_0 + \alpha \sum_{j=1}^{i} y_j (1-\alpha)^{i-j}, \quad i = \overline{1,m}$$

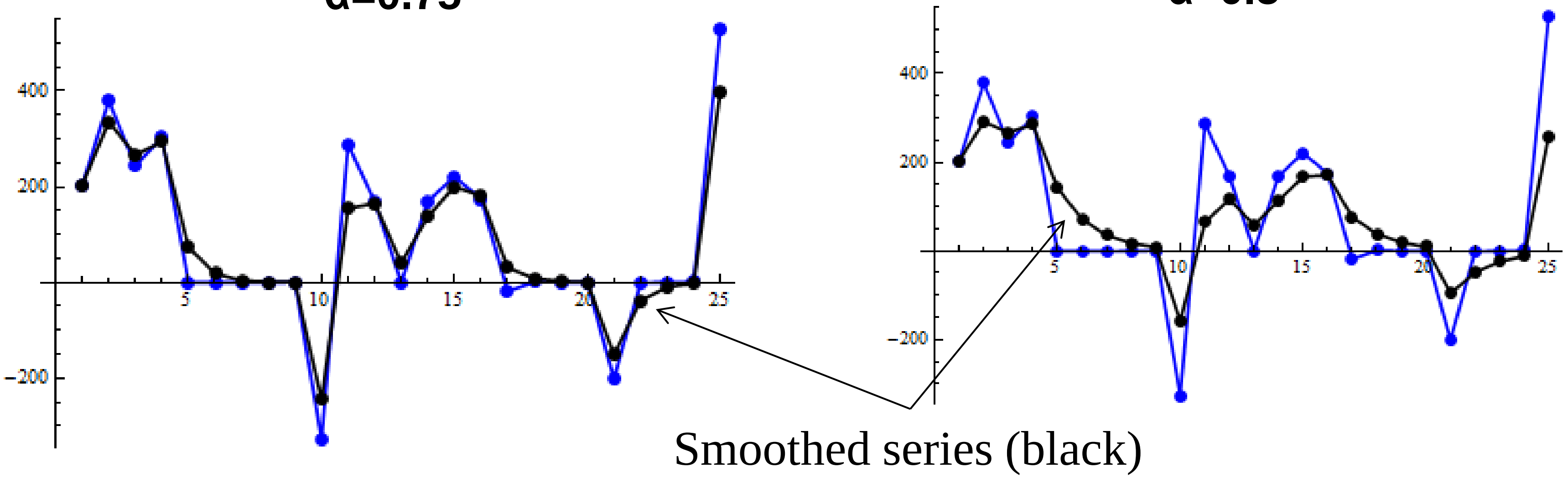
Remarks:

- If $\alpha=1$ then there is no smoothing; if $\alpha=0$ then the entire series is smoothed (it will have the value of the first element)
- Exponential smoothing considers that the most recent points are more important; the influence of previous values can be controlled by the value of α

Time series pre-processing

Example (exponential smoothing)

α=0.75 **α=0.5**



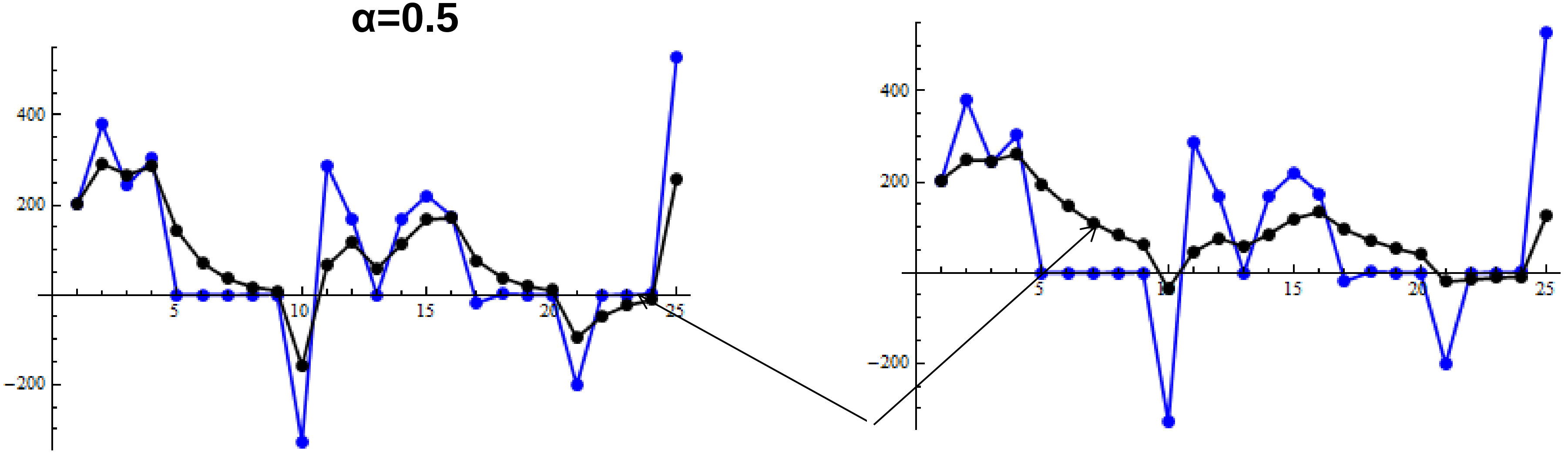
Remarks:

- If α=1 then there is no smoothing; if α=0 then the entire series is smoothed (it will have the value of the first element)
- Exponential smoothing considers that the most recent points are more important; the influence of previous values can be controlled by the value of α

Time series pre-processing

Example (exponential smoothing)

α=0.5 **α=0.25**



Remarks:

Smoothed series (black)

- If α=1 then there is no smoothing; if α=0 then the entire series is smoothed (it will have the value of the first element)
- Exponential smoothing considers that the most recent points are more important; the influence of previous values can be controlled by the value of α

Time series pre-processing

Normalization – is useful especially when multiple series are analyzed (e.g. temperature, pressure etc)

Approaches:

Range - based normalization

$$z_i = \frac{y_i - \min(y)}{\max(y) - \min(y)}, \quad i = \overline{1,m}$$

Standardization

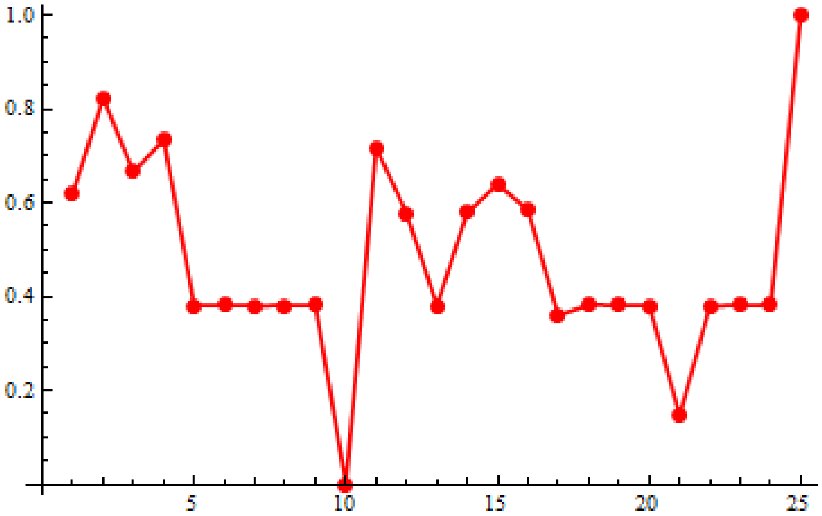
$$z_i = \frac{y_i - mean(y)}{stdev(y)}, \quad i = \overline{1,m}$$

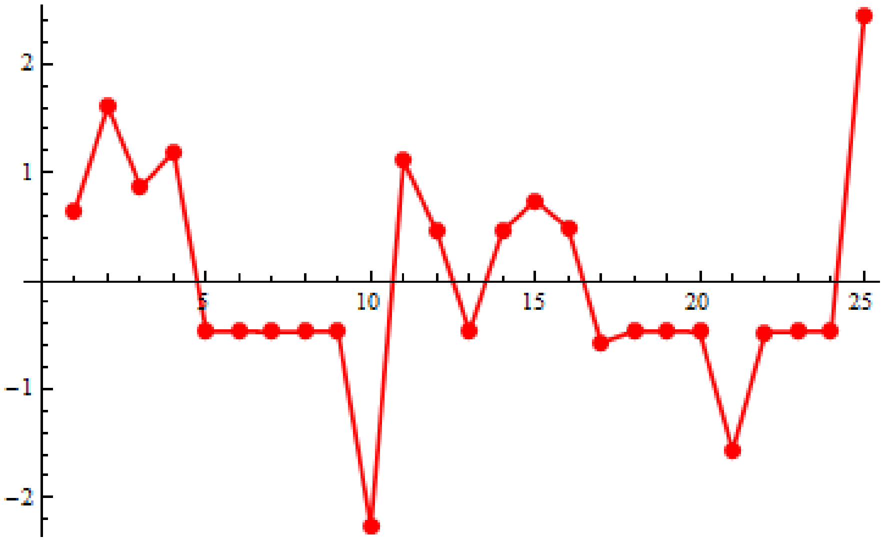
Remark:

- min(y) and max(y) are the minimal and the maximal value of the series, respectively
- mean(y) and stdev(y) are the average and the standard deviation of the series, respectively

Time series pre-processing

Example





Remark:

- Normalization and standardization preserves the shape of the series but change the range of values

Time series forecasting

Motivation:

- Predict future trends in retail sales, economic indicators, weather forecasting, stock markets

Forecasting:

- Input: one or more series
- Output: future values of the series

How can be approached:

- Interpreting it as a regression problem – estimating the explicit dependence between the behavioral attribute and time

- By using models expressing the relationship between future and previous values in the series (autoregressive models)

Time series forecasting

Remark: forecasting models “works” well for stationary series

Intuitively, a stationary time series is characterized by the fact that its statistical properties (mean, variance, autocorrelation) are constant in time

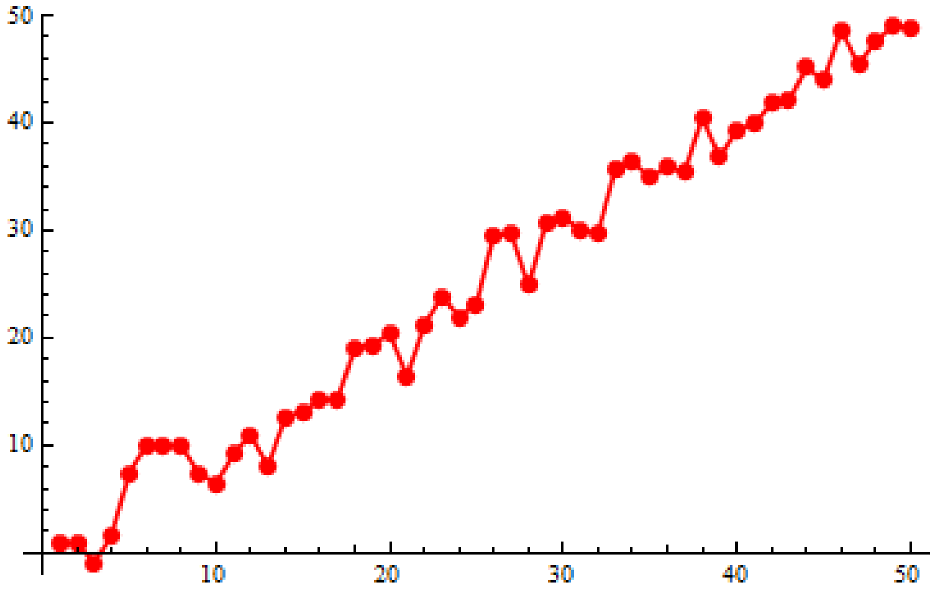
Strict stationarity: the probability distribution of the values in any time interval [a,b] is identical to the probability distribution of the values in any shifted interval [a+h, b+h] (for an arbitrary h>0)

Remark:

- this means that the window-based statistical properties of a stationary series can be estimated in a meaningful way as they do not vary over different time windows
- In the case of nonstationary series this is no more true, thus it would be useful to transform a nonstationary in a stationary series before the forecasting process

Time series forecasting

Example: a synthetic time series: y_t=t+noise (the noise is generated using a normal distribution with mean 0 and standard deviation 2)



Time series forecasting

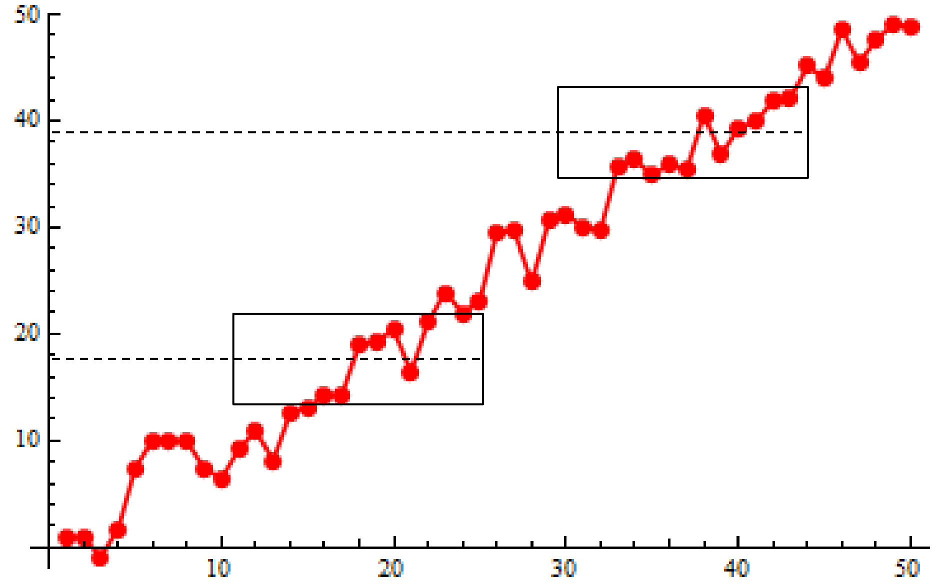
Example: a synthetic time series: y_t=t+noise (the noise is generated using a normal distribution with mean 0 and standard deviation 2)

Remark:

- this is a non-stationary time series since the averages of the values corresponding to two time-windows will be different

Average 2
(second window)

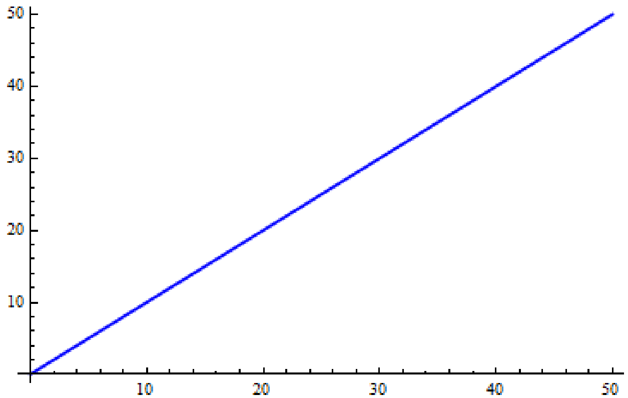
Average 1
(first window)

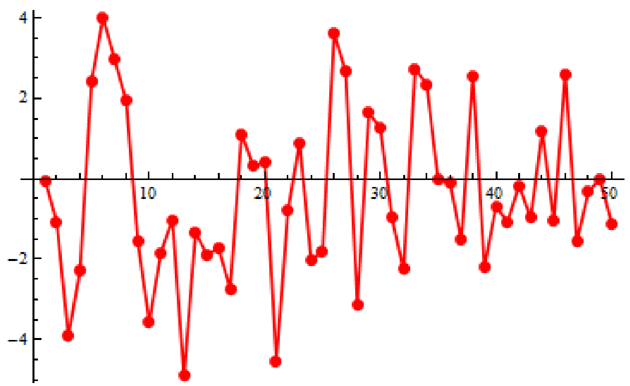


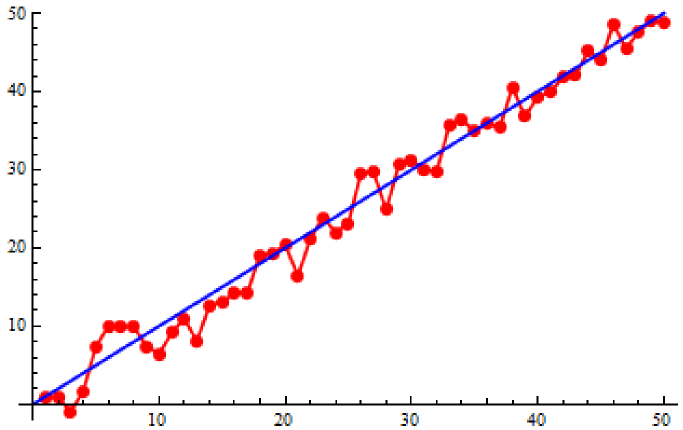
Time series forecasting

Example: a synthetic time series: y_i=i+noise (the noise is generated using a normal distribution with mean 0 and standard deviation 2)

Remark. there are two components: trend and noise





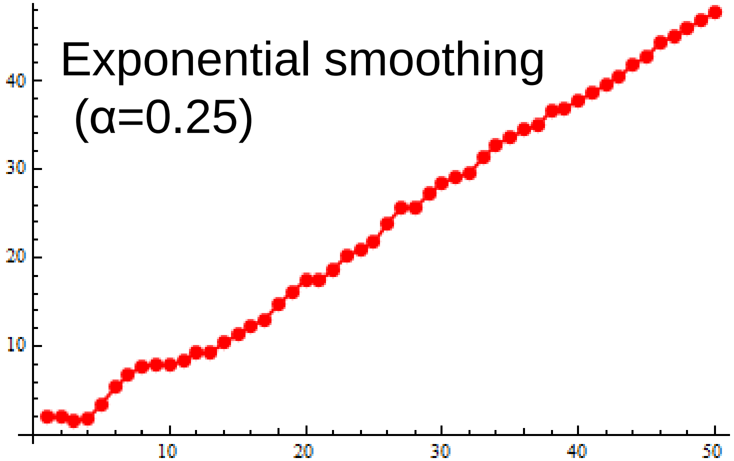


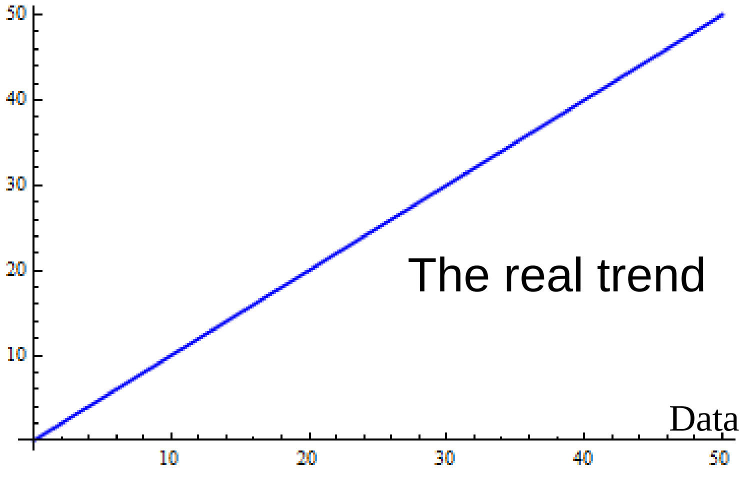
How can be extracted the two components form the initial series?

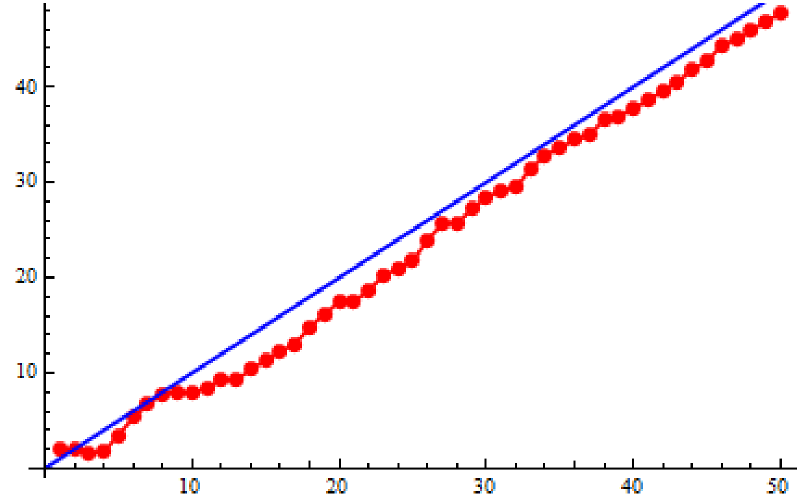
Time series forecasting

What: Extracting the trend = remove the noise

How: by smoothing





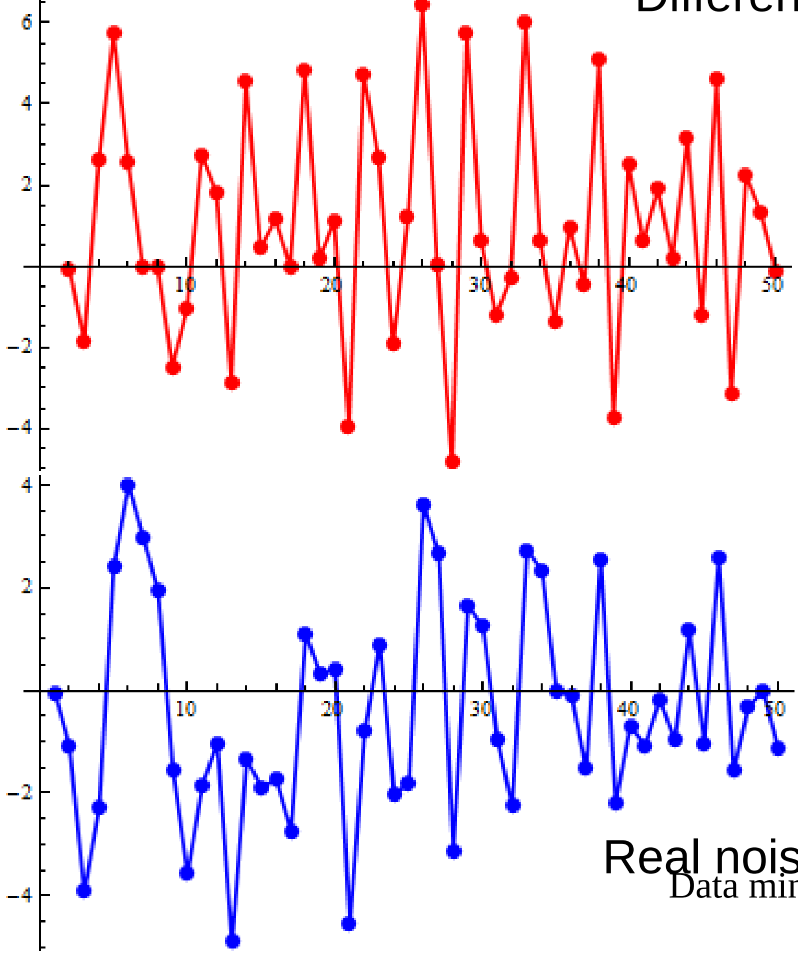


Time series forecasting

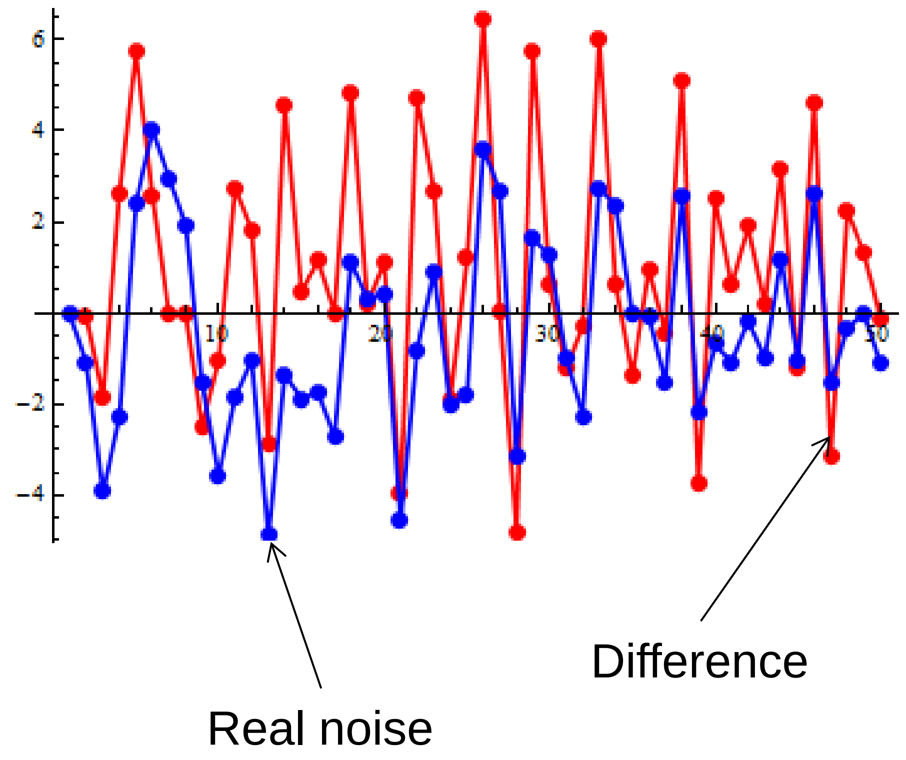
What: Extracting the noise = remove the trend

How: by differencing

Difference



Real noise



Time series forecasting

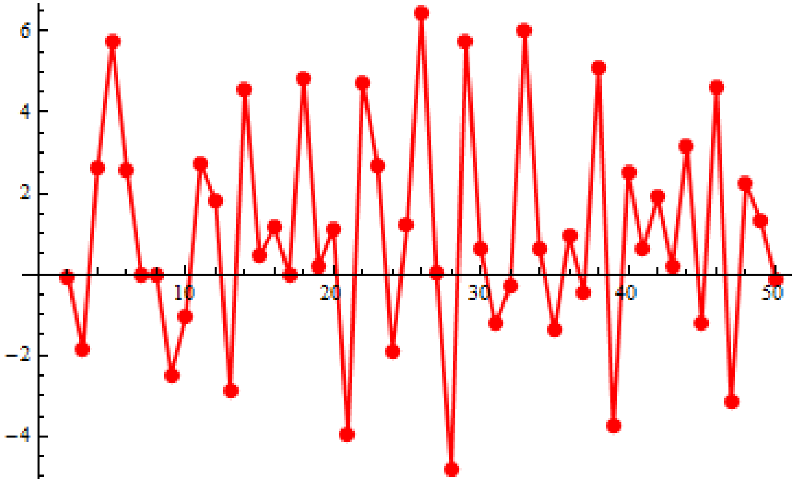
What: Extracting the noise = remove the trend

How: by differencing

The transformation by differencing: $$z_t = y_t - y_{t-1}$$

Remarks:

- The transformed series is stationary
- If there is some "seasonal" effect in data then it is better to use a lag higher than 1 (e.g. for a weekly induced effect the lag might be 7)
- In the case of geometrically increasing series (prices which increase with an almost constant inflation factor) it would be useful to apply first the logarithm on data

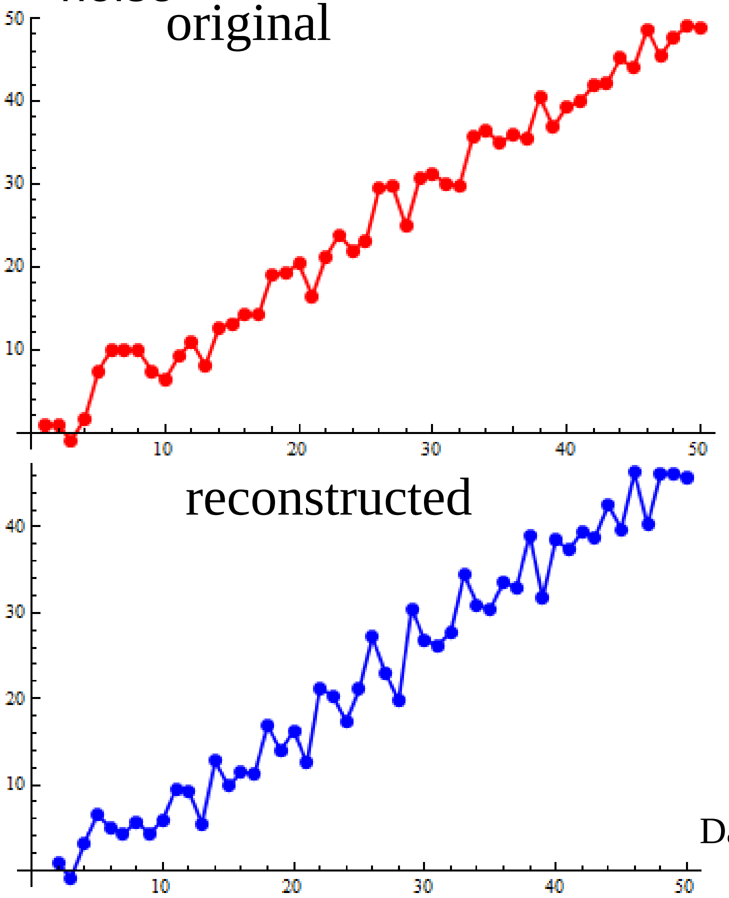


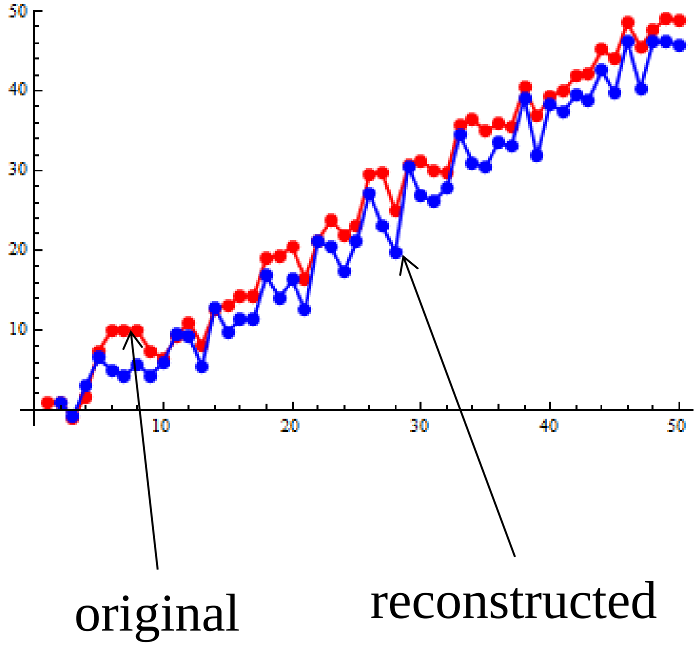
Difference

Trend and noise

Question: Can the original data be reconstructed from the trend and noise estimation?

Reconstruction: adding the estimation of the trend and the estimation of the noise





Forecasting

How can we estimate a new value of the series:

- Estimate a new value according to the trend model
- Generate a new value according to the noise model
- Add these values

Main problem:

- It is necessary to construct a model of the trend (e.g. by regression)
- It is necessary to infer the noise model; if it is a white noise (the values corresponding to different time moments are generated by independent variables with normal distribution with mean 0) then the parameters (mean and standard deviation can be easily estimated)

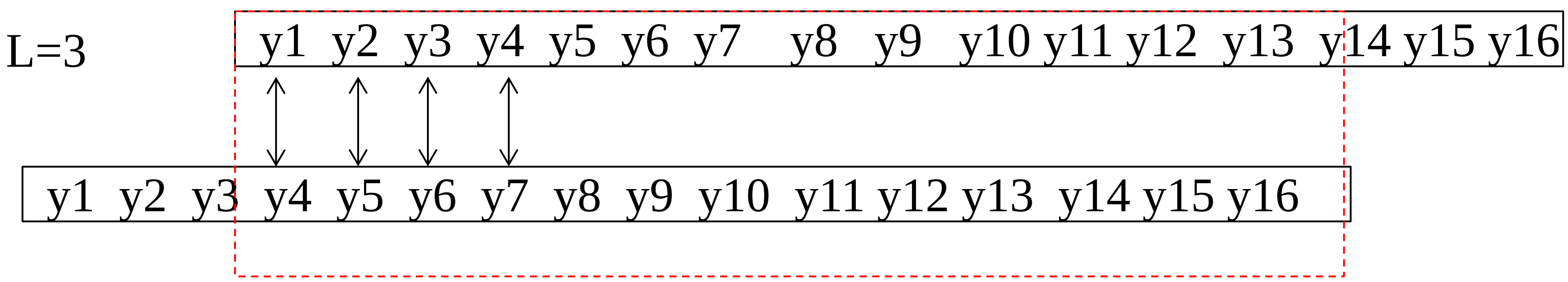
Other approach: use the autocorrelation = correlation between values corresponding to neighboring time moments

Autoregressive models

Main idea: if the autocorrelation of the time series has large absolute values then the value at a given position in the series can be predicted based on the values in the neighborhood

Computation of autocorrelation of a stationary series, $(y_1,y_2,\ldots,y_n)$, as correlation (Pearson correlation coefficient) between values separated by a given time-lag, L

$$Autocorrelation(L) = \frac{\frac{1}{n-L}\sum_{i=1}^{n-L}(y_i - avg(Y))(y_{i+L} - avg(Y))}{\mathrm{var}(Y)}$$

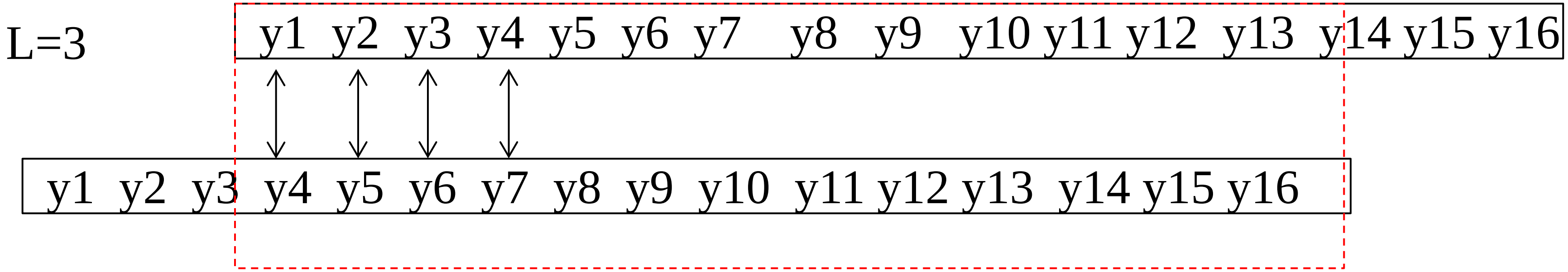


Autoregressive models

Remarks:

- The autocorrelation belongs to [-1,1]
- For small values of the lag L the autocorrelation value is most probably positive
- High absolute values of the autocorrelation suggest that the value at a given position in the series can be predicted as a function of the values in the neighboring window

$$Autocorrelation(L) = \frac{\frac{1}{n-L}\sum_{i=1}^{n-L}(y_i - avg(Y))(y_{i+L} - avg(Y))}{\mathrm{var}(Y)}$$



Autoregressive models

General form of a simple autoregressive model of order p: AR(p)

$$y_t = \sum_{i=1}^{p} a_i y_{t-i} + c + \varepsilon_t$$

Remarks:

- p is the order of the model and it should be chosen by analyzing various values of the time-lag L; p should be chosen as the first value of L (starting the search from L=1) for which the absolute value of the auto-correlation becomes small enough
- a_1, a_2,…., a_p and c are parameters of the model and should be estimated by using some training data by least squares regression
- ε_t denotes the noise

Autoregressive models

Moving Average models : MA(q)

Motivation:

- The simple autoregressive models are not able to explain all variations (particularly sudden changes, e.g. shocks)

Idea:

- The moving average models predict further values based on previous deviations from predicted values

$$y_t = \sum_{i=1}^{q} b_i \varepsilon_{t-i} + c + \varepsilon_t$$

Remark:

- Under the assumption that the time series is stationary and the noise has zero mean the value of c is the average of the time series
- The parameters $b_1, b_2, \ldots, b_q$ should be estimated from data (based on solving a nonlinear fitting problem)

Autoregressive models

Autoregressive Moving Average models : ARMA(p,q)

Motivation:

- Combine the power of the autoregressive and moving average models

$$y_t = \sum_{i=1}^{p} a_i y_{t-i} + \sum_{i=1}^{q} b_i \varepsilon_{t-i} + c + \varepsilon_t$$

Remarks:

- An important aspect is the choice of p and q: they should be the smallest values which ensure a good fit of data – not easy to find
- Non-stationary timeseries: combine differencing with ARMA → ARIMA (Autoregressive Integrated Moving Average) ARIMA(p,d,q)

$$y'_t = \sum_{i=1}^{p} a_i y'_{t-i} + \sum_{i=1}^{q} b_i \varepsilon_{t-i} + c + \varepsilon_t$$

$$y'_t = y_t - y_{t-d}$$

Motif discovery

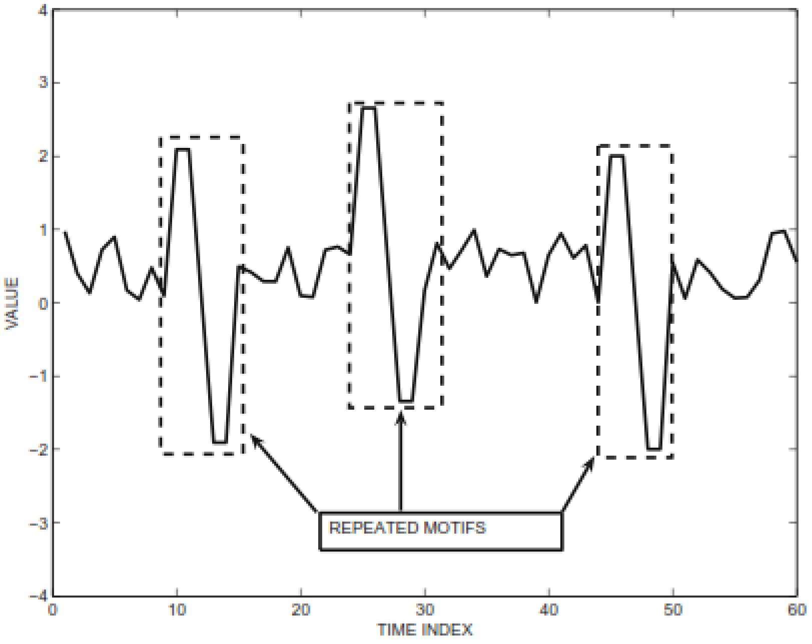
Motif = frequently occurring pattern in the time series

Pattern = sequence of values which usually correspond to consecutive positions in a time series

Discovery process – it requires:

- At least one time series
- The length L of the motif
- A similarity/dissimilarity measure
- A threshold for dissimilarity

Output: a subsequence of length L which is most frequent in the timeseries



C. Aggarwal, Data Mining – the Textbook, 2015

Motif discovery

Motif = frequently occurring pattern in the time series

Example: Brute force algorithm ($O(n^2)$ distance computations)

```
FindMotif (y[1..n],L,eps)
countMax=0
FOR i=1,n-L+1 DO
    candidate=y[i..i+L-1]
    count=0
    FOR j=1,n-L+1 DO
        D=dist(y[i..i+L-1],y[j..j+L-1])
        IF (i!=j) and (D<=eps) THEN count=count+1
    ENDFOR
    IF count[i]>countMax THEN best=i; countMax=count
  ENDFOR
  RETURN (y[best..best+L-1])
```

Outliers

There can be two main types of outliers in time series:

Point outliers:

- A significant deviation from expected (forecasted) values
- For instance, a sudden change in the time series values

Shape outliers:

- A consecutive pattern of data points might be an anomaly even is the individual values are not necessary unusual
- For instance, in a ECG series an irregular heart beat may be considered an anomaly

Outliers

Point outliers detection:

Step 1: determine the forecasted values of the time series at each timestamp ($z_m, z_{m+1}, \ldots, z_n$)

Step 2: construct the time series of deviations ($d_m, d_{m+1}, \ldots, d_n$) with $d_i = z_i - y_i$

Step 3: compute the standardized deviations ($s_m, s_{m+1}, \ldots, s_n$) with

$s_i = (d_i - avg(d))/stdev(d)$

If the absolute value of s_i is larger than a given threshold (e.g. 3) then at time I it could be an anomalous value

Outliers

Shape outliers detection:

Step 1: Extract all subseries corresponding to a window of size W

Step 2: Compute the distance between each subseries and all the other subseries corresponding to non-overlapping windows

Step 3: The subseries significantly different with respect to the other are considered potential outliers

Issues:

- Choice of W
- Choice of the threshold