UNIVERSITY OF MARYLAND

MSML610: Advanced Machine Learning

Time Series Machine Learning

Instructor: GP Saggese, PhD - `gsaggese@umd.edu`

References:

Time Series

- **Time Series**
  - Basic definition
  - Time series operators
  - Time series decomposition
- Classical Methods
- Modern Approaches
- Special techniques for time series modeling

Basic definition

- Time Series
  - **Basic definition**
  - Time series operators
  - Time series decomposition
- Classical Methods
- Modern Approaches
- Special techniques for time series modeling

Time Series

- A **time series** is a sequence of observations over time, e.g.,
  - Finance: Hourly stock prices
  - Web Analytics: Number of active users on a site sampled every minute
  - Manufacturing: Sensor data from machinery (e.g., temperature or vibration)
  - Weather: Daily temperature measurements
  - Energy: Daily electricity usage of a household
- You need time series only for things that change over time
  - Everything in the real world (besides mathematical objects) changes over time
  - You need time series for pretty much everything!

Time Series

- A time series is modeled as a random process, a sequence of random variables indexed by time:
$$\{Y_t\}_{t=-\infty}^{\infty}$$
    - Variables can be continuous or discrete
    - Data is typically equi-spaced in time
    - The time dimension matters since random variables exhibit dependence over time
- **Goals:**
    - Understand patterns
        - Identify trends, seasonality, and cyclic behavior
        - Detect anomalies or outliers
    - Predict future values
        - Forecast future data points based on historical patterns
        - E.g., predicting next month's sales based on past sales data
    - Improve decision-making
        - Use insights from the time series to make informed business or policy decisions
        - E.g., adjusting inventory levels based on predicted demand

Time Series: Visualization and Exploration

- **Visualization:**
  - Guides preprocessing choices
  - Helps form hypotheses before modeling
  - Distinguishes between underlying structure and randomness
- **Patterns:**
  - **Trend**: Long-term increase or decrease
  - **Seasonality**: Repeating patterns at regular intervals
  - **Noise**: Random fluctuations
- **Line plots** show raw data over time, e.g.,
  - Trend presence
  - Outliers or abrupt changes
- **Seasonal plots** reveal periodic patterns
  - E.g., plot monthly sales to find yearly seasonality
- **Autocorrelation plots** detect repeating structures

Autocovariance

- The j-lag **autocovariance** of a time series $\{Y_t\}$ is the covariance of a random variable and the variable j samples before:

$$\mathrm{Cov}(Y_t, Y_{t-j}) \stackrel{def}{=} \mathbb{E}[(Y_t - \mathbb{E}[Y_t])(Y_{t-j} - \mathbb{E}[Y_{t-j}])]$$

  - Represent how a variable varies over time
  - Linear relationship
- The j-lag **autocorrelation** of a time series $\{Y_t\}$ is:

$$\mathrm{Corr}(Y_t, Y_{t-j}) = \rho(Y_t, Y_{t-j}) \stackrel{def}{=} \frac{\mathrm{Cov}(Y_t, Y_{t-j})}{\sqrt{\mathbb{V}[Y_t]\mathbb{V}[Y_{t-j}]}}$$

  - Measure strength and direction of the linear relationship between samples
  - Scale-free

Stationarity: Intuition

- A time series $\{Y_t\}$ is **stationary** if some statistical properties (e.g., mean, variance, autocorrelation structure) do not change over time
  - I.e., the time series is "unchanged" by shifts in time
  - Stationarity is analogous to IID sampling for random variables
- Time series are rarely stationary
  - Stationarity is often an approximation/simplification of reality
- **Why important:**
  - Many models (e.g., ARIMA) assume stationarity
  - E.g., raw stock prices are non-stationary, returns often are
- **Tests for stationarity:**
  - ADF Test (Augmented Dickey-Fuller): tests for unit root
  - KPSS Test: tests for trend stationarity

Strictly Stationary: Definition

- A time series $\{Y_t\}$ is **strictly stationary** iff for any any set of $r > 0$ indices $t_1, t_2, ..., t_r < t$, the joint distribution of $(Y_{t_1}, Y_{t_2}, ..., Y_{t_r})$ depends only on the differences $t_1 - t_2, ..., t_1 - t_r$
- E.g.,
    - (Y_1, Y_5) has the same joint distribution as (Y_{12}, Y_{16})
    - (Y_1, Y_2, Y_3) has the same joint distribution as (Y_3, Y_4, Y_5)
- **Intuition**:
    - The data (i.e., joint probability of any set of observations) is invariant when we shift it in time
    - Only the distances in time matter
- If $\{Y_t\}$ is strictly stationary:
    - All moments (e.g., mean, variance) of Y_t don't depend on t
    - Any statistics between lags of the time series depend only on the difference in time between lags

Weakly Stationary: Definition

- **Weakly stationarity** requires milder assumptions than for strictly stationary process:

  1. The mean is constant over time:
  $$\mathbb{E}[Y_t] = \mu \ \forall t$$
  2. The variance is constant over time:
  $$\mathbb{V}[Y_t] = \sigma^2 \ \forall t$$
  3. The j-lag autocovariance $\text{Cov}(Y_t, Y_{t-j})$ depends on distance between lags j but not on t:
  $$\text{Cov}(Y_t, Y_{t-j}) = \gamma_j$$

- In practice, there is a constraint only on:
  - First and second moments
  - The joint distribution of 2 time indices

- **Intuition**:
  - No trend (mean is constant)
  - Variations around the mean have constant amplitude (variance is constant)
  - Consistent wiggling (random patterns look the same)

Auto-Correlation Function (ACF)

- **Auto-correlation function** (ACF) is a graphical representation of the i-lag autocorrelation of a time series
  - Plot the correlation coefficient of a time series with its own lagged values
  - (Ideally) Plot the uncertainty of the coefficients
  - Helps identify repetitive patterns or seasonality in the data
  - E.g., for a time series of daily temperatures, the ACF can help you see if today's temperature is correlated with the temperature from previous days
- **Partial Auto-Correlation Function** (PACF) is like ACF but controls for the values of the time series at all shorter lags

$$\alpha(k) = \mathrm{Corr}(Y_t - Proj_{t,k}(Y_t), Y_{t-k} - Proj_{t,k}(Y_{t,k}))$$

where:

  - $Proj_{t,k}(x)$ is the projection of x onto the space spanned by $(x_t, ..., x_{t-k+1})$
  - Helps understand the direct relationship between a time series and its lagged values, excluding the influence of intermediate lags
  - E.g., in a sales data time series, PACF can help identify the direct impact of sales from a specific previous month on the current month's sales, without the influence of other months in between

Transformation of a time series

- Any deterministic transformation $g(\cdot)$ of a strictly (weakly) stationary process $\{Y_t\}$ is also strictly (weakly) stationary
- **Examples**:
  - Log transformations useful when data grows exponentially
  - Differencing removes trend and makes series stationary
    - First difference: $z_t = y_t - y_{t-1}$
  - Power transformations (e.g., square root) can reduce skewness
  - Detrending
    - Subtract a fitted trend line
    - Apply moving average smoothing
- A transformation can make a process stationary, e.g.,
  - Detrending
  - Differencing (integral or fractional)

Time series operators

- Time Series
  - Basic definition
  - **Time series operators**
  - Time series decomposition
- Classical Methods
- Modern Approaches
- Special techniques for time series modeling

Lag Operator

- Aka "shift back", backshift, delay
- Given a time series $\{X_t\}$, the lag operator $L(\cdot)$ generates another time series:

$$Y_t = LX_t = X_{t-1}$$

- **Intuition**
  - Lagging a time series means that the t (today) element of the new time series is the $t-1$ (yesterday) element of the old time series
  - It delays the time series
- The "normal" direction (i.e., positive delay) is delaying, since you are not snooping in the future
  - `pd.shift(n>0)` with a positive sign

```
date        val    val.shift(2)
2016-03-10  0      nan
2016-03-11  1      nan
2016-03-14  2      0
2016-03-15  3      1
2016-03-16  4      2
```

  - The values at the beginning of the period are not available since they require data before the period of interest

Lead Operator

- Aka "shift forward"
- It is accomplished by:

$$Y_t = L^{-1} X_t = X_{t+1}$$

  - When using a variable function of time, the transformation is like $x(t+2)$ since the value today $x(0)$ is the value computed in the future $x(2)$

- When you shift forward (lead) `df.shift(n<0)`, you move a value from the future (a value computed n periods in the future) to today

```
date        val     val.shift(-2)
2016-03-10  0       2
2016-03-11  1       3
2016-03-14  2       4
2016-03-15  3       nan
2016-03-16  4       nan
```

- This is equivalent to "shifting up" a time series ordered by increasing dates
  - Some values at the end of the period won't be available since they would have been computed after the period of interest is over
  - Some values computed at the beginning of the period will be discarded

Shifting More Than One Time Step

- You can shift more than one lag with:

$$L^k X_t = X_{t-k}$$

$$L^{-k} X_t = X_{t+k}$$

- Important points:
  - L^k represents a backward shift by k time steps
  - L^{-k} represents a forward shift by k time steps
- E.g., if X_t is a time series:
  - $L^2 X_t = X_{t-2}$ shifts the data point two steps back
  - $L^{-3} X_t = X_{t+3}$ shifts the data point three steps forward

Difference Operator

- The first difference of a time series is defined as the time series:

$$Y_t = \Delta X_t \stackrel{def}{=} X_t - X_{t-1}$$

  i.e., the time series that is the difference between the original time series and its lagged version

- The first difference can be written in terms of lag operator as:

$$\Delta X_t = (1 - L)X_t$$

- **Intuition**:
  - Differencing means computing the difference between consecutive observations:

$$Y_t = X_t - X_{t-1}$$

  - This means removing the changes in the level of a time series
    - Eliminate trend and seasonality, which stabilize the mean of the time series
    - Differencing can make a time series stationary

Second Difference Operator

- The second difference is defined as:

$$\Delta^2 X_t = \Delta(\Delta X_t)$$

- Note that the second difference is not the difference $Y_t - Y_{t-2}$
  - Developing:

$$Y_t = \Delta X_t = X_t - X_{t-1}$$

  and

$$\begin{aligned} Z_t &= Y_t - Y_{t-1} \\ &= X_t - X_{t-1} - (X_{t-1} - X_{t-2}) \\ &= X_t - 2X_{t-1} + X_{t-2} \end{aligned}$$

- The second difference operator can be written in terms of lag operator as:

$$\Delta^2 X_t = (1 - L)^2 X_t$$

N-th Difference Operator

- The n-th difference operator is defined:

$$\Delta^n X_t = (1 - L)^n X_t$$

- The notation highlights that you can express the n-th difference operator in terms of n-th power of the difference operator

Time series decomposition

- Time Series
  - Basic definition
  - Time series operators
  - **Time series decomposition**
- Classical Methods
- Modern Approaches
- Special techniques for time series modeling

Decomposition of Time Series

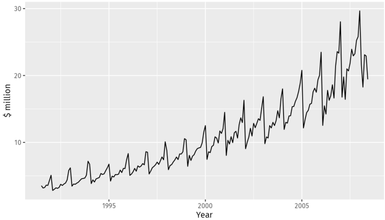
- A time series can be broken into several components:

$$y_t = \text{Trend}_t + \text{Seasonality}_t + \text{Cycle}_t + \text{Residual}_t$$

  - **Trend**
    - Long-term increase or decrease in data
    - Can change direction over time
  - **Seasonality**
    - Affected by seasonal factors (e.g., time of day, day of week, month of year)
    - Fixed and known frequency
  - **Cycle**
    - Value rises and falls without fixed frequency
    - E.g., economic conditions exhibit cycles
  - **Residual (noise)**
    - Everything that is left after removing the other components
    - Idiosyncratic component
- Besides additive model, the components can also be mixed in different ways (e.g., multiplicative)

Seasonality: Example

- Consider sales of antidiabetic drug over time
  - Sharp spike in January
  - Dip in February
  - Increase over the year
- **Why?**
  - In January, government subsidy makes it cost-effective to stockpile drugs
  - In February, dip occurs as people have already bought many drugs
  - Demand increases until December as people use their reserves
  - Then the cycle repeats next year

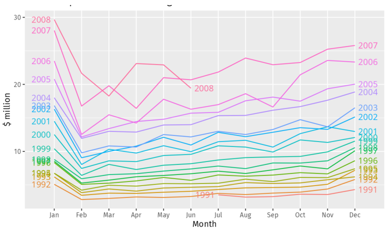


Cycle: Example

- Gross Domestic Product (GDP) moves around its long-term growth trend
  - The fluctuation is the business cycle
  - Understanding these cycles aids in economic planning and forecasting
- Different cycles:
  - **Inventory**: 3-5 years
    - Driven by inventory level changes in response to demand fluctuations
    - E.g., a retailer overstocks during high demand and then reduce orders, leading to a cycle
  - **Fixed investment**: 7-11 years
    - Investments in long-term assets like machinery and buildings
    - E.g., a manufacturing company invests in new machinery every decade to improve efficiency
  - **Infrastructural investment**: 15-25 years
    - Large-scale infrastructure projects such as roads, bridges, and public utilities
    - E.g., a government plans a major highway expansion every 20 years to accommodate growing traffic
  - **Technological investment**: 45-60 years
    - Major technological advancements and their adoption across industries
    - E.g., the widespread adoption of the Internet in 1995-2010

Seasonal Plot

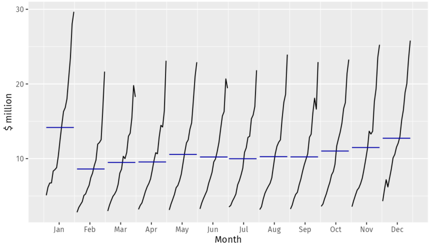
- Seasonal plot allows understanding the model structure over time
- Assume you know the periodicity of a signal
  - E.g., yearly periodicity of a monthly time series
- Partition the time-series based on the periodicity:
  - Plot each time series chunk on the same graph



- **Questions:**
  - Do the data exhibit a seasonal pattern?
  - Is there a within-group pattern (e.g., Jan and July exhibit similar patterns)?
  - Are there outliers after accounting for seasonality?
  - Is the seasonality changing over time?

Seasonal Differencing

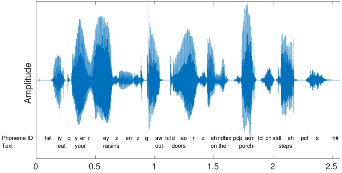
- Take the difference between observations at the same point of consecutive periods
    - E.g., for time series with yearly periodicity, take the Year-over-Year (YoY) difference
- Remove seasonal effects in time series data with strong seasonal patterns (e.g., retail sales or temperature data)
    - Help identify underlying trends by eliminating seasonal fluctuations

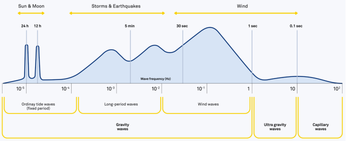


Sub-seasonal plot

Spectral Plot

- Spectral plot estimates spectral density of a process from time samples of the signal
    - Detect periodicity
    - Identify dominant frequencies
    - Analyze power distribution over frequency
- E.g., a spectral plot identifies different frequencies in a sound recording
    - Noise reduction or enhancement of certain frequencies
    - Speech recognition
- E.g., compute the frequency of ocean waves





Classical Methods

- Time Series
- **Classical Methods**
  - Simple Models for Stochastic Process
  - Autoregressive models
  - Moving average models
  - ARMA(p, q) process
  - ARCH model
- Modern Approaches
- Special techniques for time series modeling

Simple Models for Stochastic Process

- Time Series
- Classical Methods
  - **Simple Models for Stochastic Process**
  - Autoregressive models
  - Moving average models
  - ARMA(p, q) process
  - ARCH model
- Modern Approaches
- Special techniques for time series modeling

White Noise Process

- Defined as:

$$\{Y_t\} \sim \text{WN}(0, \sigma^2)$$

- Each Y_t is:
  - A IID random variable at time t
    - Independent over time
    - Drawn from the same distribution $Y_t \sim$ IID from distribution F (not necessarily Gaussian)
  - With mean $\mathbb{E}[Y_t] = 0$ and constant variance $\mathbb{V}[Y_t] = \sigma^2$
- **Key points:**
  - It's strictly stationary
  - $\{Y_t\}$ is uncorrelated over time
  - Variance σ^2 is constant for all t
  - $\text{Cov}(Y_t, Y_{t-j}) = \gamma_j = 0$ for $j \neq 0$
- It's called "white noise" because:
  - There is no pattern ("noise")
  - Signal spectrum contains all frequencies with equal power (analogous to "white light")

Deterministically Trending Process

- Defined as:

$$Y_t = \beta_0 + \beta_1 t + \varepsilon_t$$

  where:
  - The noise is Gaussian: $\varepsilon_t \sim \text{GWN}(0, \sigma_\varepsilon^2)$
  - The noise term is also called "innovation" or "error term"
  - β_0 is the intercept of the model
  - β_1 is the slope of the model, representing the trend over time
  - t is the time index
- The mean $\mathbb{E}[Y_t] = \beta_0 + \beta_1 t$ depends on t
  - It is non-stationary in the mean due to the presence of the trend component $\beta_1 t$, e.g.,
  - $\beta_1 > 0$ indicates an upward trend
  - $\beta_1 < 0$ indicates a downward trend
- The variance of the noise term σ_ε^2 is constant over time

Random Walk

- Defined as
$$Y_t = Y_{t-1} + \varepsilon_t$$
where
  - The noise is Gaussian: $\varepsilon_t \sim \text{GWN}(0, \sigma_\varepsilon^2)$
- It can be rewritten in terms of the noise terms doing a recursive substitution:
$$Y_t = Y_0 + \sum_{i=1}^{t} \varepsilon_i$$
- The mean is constant: $\mathbb{E}[Y_t] = \mathbb{E}[Y_0] = \mu$
- The variance is $\mathbb{V}[Y_t] = t\sigma_\varepsilon^2$ since all the covariances between innovations are 0
  - It is non-stationary in the variance

Autoregressive models

- Time Series
- Classical Methods
  - Simple Models for Stochastic Process
  - **Autoregressive models**
  - Moving average models
  - ARMA(p, q) process
  - ARCH model
- Modern Approaches
- Special techniques for time series modeling

Autoregressive (AR) Models

- An AR model of order p expresses future values using past p values:

$$
\begin{aligned}
y_t &= c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t \\
&= c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t
\end{aligned}
$$

  where:
  - y_t is the value at time t
  - c is a constant
  - $\phi_1, \phi_2, \ldots, \phi_p$ are the model parameters
  - $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$ are past values (aka lags)
  - Random error term is white noise $\varepsilon_t \sim$ IID $Normal(0, \sigma^2)$
- E.g., predicting temperature today using temperature for past 3 days:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \phi_3 y_{t-3} + \varepsilon_t$$

AR(1) process

- Aka "auto-regressive of order 1"
- AR(1) model is defined as:

$$Y_t = c + \phi Y_{t-1} + \varepsilon_t$$

  where
  - the model is an autoregressive term plus noise
  - the noise is IID Gaussian $\varepsilon_t \sim \text{GWN}(0, \sigma_\varepsilon^2)$
- It is regressive with respect to itself, i.e., "auto-regressive"
  - The representation:

$$Y_t = \phi Y_{t-1} + \varepsilon_t$$

    can be thought as a regression of Y_t against Y_{t-1}

AR(1) process: mean

- Applying the expected value to the definition of AR(1)

$$Y_t = c + \phi Y_{t-1} + \varepsilon_t$$

  you get:

$$\mathbb{E}[Y_t] = c + \phi \mathbb{E}[Y_{t-1}] + \mathbb{E}[\varepsilon_t]$$

- Assuming the mean exists and it's μ, then $\mu = c + \phi\mu$ so:

$$\mathbb{E}[Y_t] = \frac{c}{1-\phi}$$

AR(1) process: in terms of mean

- Rewriting the AR(1) model:

$$Y_t = c + \phi Y_{t-1} + \varepsilon_t$$

using the relationship for the mean:

$$\mu = \frac{c}{1-\phi}$$

you get:

$$Y_t = \mu(1-\phi) + \phi Y_{t-1} + \varepsilon_t$$

- Rewriting in terms of difference from the mean:

$$(Y_t - \mu) = \phi \cdot (Y_{t-1} - \mu) + \varepsilon_t$$

  - It is like random walk $Y_t = Y_{t-1} + \varepsilon_t$ but with a mean μ and a param ϕ

AR(1) process: properties

- Using the definition of model in terms of the mean:

$$Y_t = \mu(1-\phi) + \phi Y_{t-1} + \varepsilon_t$$

you can compute the statistical properties of AR(1) process:

$$\begin{aligned}
&\mathbb{E}[Y_t] = \mu \\
&\mathbb{V}[Y_t] = \frac{\sigma_\varepsilon^2}{1-\phi^2} \\
&\mathrm{Cov}[Y_t, Y_{t-j}] = \mathbb{V}[Y_t]\phi^j \\
&\rho(Y_t, Y_{t-j}) = \phi^j
\end{aligned}$$

- If $-1 < \phi < 1$, the AR(1) model is weakly stationary

AR(1) process is mean-reverting

- Mean-reverting means that when the value is far from the mean, it tends to go back
  - The speed of mean reversion depends on ϕ

AR(1) Process Approximates Ergodicity

- Ergodicity is a property of time series where Y_t and Y_{t-j} become independent as j grows large enough
  - "Memory" fades over time
- The autocorrelation of AR(1) has a geometric decay:

$$\text{Cov}[Y_t, Y_{t-j}] = \mathbb{V}[Y_t]\phi^j$$

  i.e., variables that are closer in time are more correlated than variables that are farther in time
  - If $j \to \infty$ then $\text{Cov}[Y_t, Y_{t-j}] \to 0$ (ergodicity)

AR(1) Process vs Gaussian White Noise

- The Gaussian White Noise (GWN) is choppy
- The AR(1) process is smoother than the GWN due to the autocorrelation in time
- AR(1) as function of ϕ:
  - $\phi = 0$: white noise, it bounces around the mean
  - $0 < \phi < 1$: it stays far from the mean and then reverts (it is smoother)
  - $\phi = 1$: random walk, it walks away from the mean
  - $\phi > 1$: explosive progress since it diverges accelerating
  - $\phi < 0$:it is super choppy

AR(1) to model financial time series

- Good model for:
  - Interest rates
  - Growth rate of macroeconomic variables (e.g., GDP, unemployment rate)
  - Profit and losses
- Bad model for:
  - Stocks don't show a strong time dependency
    - Returns look like white noise
    - Prices look like random walk

AR(p) model

- AR(p) is an autoregressive model of order p:

$$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t$$

  where

  - ε terms are white noise

- In words, AR(p) models are linear combination of p past realization plus noise
- AR models assume the time series is stationary:
  - Partial Autocorrelation Function helps choose p
  - Model parameters are estimated using methods like Ordinary Least Squares (OLS) or Maximum Likelihood Estimation (MLE)

AR(p) model in terms of lag operator

- The AR(p) equation is:

$$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t$$

- Separating var and noise term

$$Y_t - \sum \phi_i Y_{t-i} = c + \varepsilon_t$$

- Using lag operator

$$\begin{aligned} Y_t - \sum \phi_i L^i Y_t = \\ (1 - \sum \phi_i L^i) Y_t = \\ \text{polynomial}(\phi, L) Y_t = c + \varepsilon_t \end{aligned}$$

Moving average models

- Time Series
- Classical Methods
  - Simple Models for Stochastic Process
  - Autoregressive models
  - **Moving average models**
  - ARMA(p, q) process
  - ARCH model
- Modern Approaches
- Special techniques for time series modeling

Moving Average (MA) Models

- A MA model of order q predicts the next value using past q errors:

$$Y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$
$$= \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

  where:
  - μ is the mean of the series
  - ε_t is the white noise error term at time t
  - $\theta_1, \theta_2, \ldots, \theta_q$ are the parameters of the model
- E.g., correcting for sensor noise by using past error patterns
  - If a sensor consistently overestimates by a small amount, the MA model can adjust for this by considering past errors
- MA models are always stationary
  - Suitable for time series data where the impact of a shock is short-lived
  - Useful for modeling time series with short-term dependencies

MA(1) process: def

- Aka "moving average of order 1"
- MA(1) model is defined as:

$$Y_t = c + \varepsilon_t + \theta\varepsilon_{t-1}$$

  where the noise is iid Gaussian: $\varepsilon_t \sim \mathrm{GWN}(0, \sigma_\varepsilon^2)$
- Why is it called moving average?
  - Consider:

$$\begin{aligned} Y_t &= c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} \\ Y_{t-1} &= c + \varepsilon_{t-1} + \theta_1\varepsilon_{t-2} + \theta_2\varepsilon_{t-3} \end{aligned}$$

  - It's like a window with given coefficients (computing an average), moving in time

MA(1) process: properties

- Using the definitions we obtain:

$$\mathbb{E}[Y_t] = c$$
$$\mathbb{V}[Y_t] = (1+\theta)\sigma_\varepsilon^2$$
$$\text{Cov}[Y_t, Y_{t-1}] = \theta\sigma_\varepsilon^2$$
$$\text{Cov}[Y_t, Y_{t-j}] = 0,\ \forall j > 1$$

- It is a weakly stationary process since:
  - Mean and variance are constant
  - The covariance depends only on the difference of the lags

MA(q) model

- MA(q) is a moving average model of order q:

$$Y_t = c + \varepsilon_t + \sum_{i=1}^{p} \theta_i \varepsilon_{t-i}$$

  where:

  - c is the mean
  - ε terms are white noise

- In words, MA(q) models are linear combination of q error terms from the past
- Intuition of covariance structure
  - In general MA(q) has dependency between consecutive terms Y_t up to Y_{t-q}
  - It can be seen by considering

$$Y_t = f(\varepsilon_t, \varepsilon_{t-1}, ..., \varepsilon_{t-q})$$
$$...$$
$$Y_{t-k} = f(\varepsilon_{t-k}, \varepsilon_{t-k-1}, ..., \varepsilon_{t-k-q})$$

    and noticing that there are common terms as long

MA(q) model in terms of lag operator

- The MA equation is:

$$Y_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

- Using the lag operator:

$$Y_t = \mu + (1 + \sum_{i=1}^{q} \theta_i L^i)\varepsilon_i = \mu + f(\theta, L)\varepsilon_i$$

ARMA(p, q) process

- Time Series
- Classical Methods
  - Simple Models for Stochastic Process
  - Autoregressive models
  - Moving average models
  - **ARMA(p, q) process**
  - ARCH model
- Modern Approaches
- Special techniques for time series modeling

ARMA(p, q) model

- It contains p autoregressive terms and q moving average terms:

$$ARMA(p,q) = AR(p) + MA(q)$$

  where:

  - AR part involves regressing the variable on its own lagged values
  - MA part models error term as a linear combination of lagged error terms

- A realization of an ARMA(p, q) process is:

$$\begin{aligned} Y_t &= AR(p) + MA(q) \\ &= (c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t) + (c + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t) \\ &= c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t \end{aligned}$$

- In terms of lag operator:

$$(1 - \sum_{i=1}^{q} \phi_i L^i) Y_t = c + (1 + \sum_{i=1}^{p} \theta_i L^i) \varepsilon_i$$

Residuals of ARMA model

- Residuals should be uncorrelated and normally distributed
- One can check the ACF of the residuals

ARIMA Models

- Consider $ARIMA(p, d, q)$ where:
  - p: number of autoregressive terms (AR)
  - d: order of differencing (I)
  - q: number of moving average terms (MA)
- From ARMA to ARIMA
  - ARMA models combine AR and MA components, using the lag operator:

$$(1 - \sum_{i=1}^{q} \phi_i L^i) Y_t = c + (1 + \sum_{i=1}^{p} \theta_i L^i) \varepsilon_i$$

  - The d-order differencing is $(1 - L)^d$ in terms of lag operator
- ARIMA models extend ARMA by including differencing:

$$\begin{cases} Z_t = (1 - L)^d Y_t \\ (1 - \sum_{i=1}^{p} \phi_i L^i) Z_t = c + (1 + \sum_{i=1}^{q} \theta_i L^i) \varepsilon_t \end{cases}$$

  - Compute the final polynomials

$$poly_{\Phi}(\phi, L)(1 - L)^d Y_t = c + poly_{\Theta}(\theta, L) \varepsilon_i$$

  - Transform back into delays

ARIMA Models: Differencing

- Differencing stabilizes the time series by handling non-stationary data
  - Over-differencing increases model complexity without improving accuracy
  - Under-differencing results in a non-stationary series
  - E.g., if a time series shows a linear trend, first-order differencing ($d = 1$) might be sufficient to achieve stationarity
- Can model and forecast a wide range of time series data
  - Setting p, d, or q to 0, ARIMA is simplified to a ARMA, AR, I, MA model
  - ARIMA(0, 0, 0)
    - $Y_t = \varepsilon_t$, which is white noise
  - ARIMA(0, 1, 0) = I(1)
    - $Y_t = Y_{t-1} + \varepsilon_t$, which is a random walk
  - ARIMA(p, 0, q) = ARMA(p, q)

SARIMA

- Seasonal ARIMA (SARIMA) extends ARIMA to handle seasonal patterns in time series data
- $SARIMA(p, d, q) \times (P, D, Q, s)$ where
  - p: order of non-seasonal AR (autoregressive) terms
  - d: number of non-seasonal differences
  - q: order of non-seasonal MA (moving average) terms
  - P: order of seasonal AR terms
  - D: number of seasonal differences
  - Q: order of seasonal MA terms
  - s: length of the seasonal cycle
    - E.g., 12 for monthly data with yearly seasonality
- It incorporates seasonal autoregressive and moving average terms, as well as seasonal differencing
  - Useful when strong periodic behavior exists
  - Seasonal differencing removes seasonal patterns: $y'_t = y_t - y_{t-s}$
  - E.g., forecasting monthly airline passenger data

Fitting ARMA / ARIMA models

- The original Box-Jenkins approach has 3 phases:
  1. Model identification / selection
     - Select p, d, q
     - Identify seasonality
     - Difference data, if necessary, to achieve stationarity
     - Check if variables are stationary
     - Use ACF, PACF to decide AR and MA components to use
  2. Parameter estimation
     - Pick coefficients to get best fit
  3. Model checking
     - Test estimated model
     - E.g., the residual should have no serial correlation and be stationary in mean and variance
     - If estimation is inadequate, go back to step 1) and attempt to build a better model

ARCH model

- Time Series
- Classical Methods
  - Simple Models for Stochastic Process
  - Autoregressive models
  - Moving average models
  - ARMA(p, q) process
  - **ARCH model**
- Modern Approaches
- Special techniques for time series modeling

ARCH: Intuition

- Auto-Regressive Conditional Heteroskedasticity (ARCH)
- ARCH is used to model time series that exhibit:
  - Time-varying volatility
  - Volatility clustering (periods of swings interspersed with periods of calm)
- Variance of error term (aka innovation) is described as a function of the value of the previous time periods error terms

$$\mathbb{V}[\varepsilon_t] = f(\varepsilon_{t-1}, ..., \varepsilon_{t-N})$$

- E.g., error variance follows an AR model

ARCH(q): definition

- The model for the error term of the time series is:

$$\varepsilon_t = \sigma_t \cdot z_t$$

where:

  - z_t is white noise process (stochastic part)
  - σ_t^2 is the time-dependent variance given by an AR(q) model:

$$\begin{aligned}\sigma_t^2 &= \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 \\ &= \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \ldots + \alpha_q \varepsilon_{t-q}^2 \\ &\text{where } \alpha_i \geq 0\end{aligned}$$

- The error variance is AR(q), i.e., a linear combination of squares of previous error term realizations

GARCH(p, q): definition

- GARCH stands for Generalized ARCH
  - The error variance follows an ARMA model
- The error term of a time series is modeled as:

$$\varepsilon_t = \sigma_t \cdot z_t$$

where:

  - z_t is white noise process (stochastic part)
  - σ_t^2 is the time-dependent variance given by an ARMA(p, q) model

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2$$

Exponential Smoothing Model

- Exponential smoothing is a time series forecasting method using weighted averages of past observations
  - More recent observations get more weight
  - Weights decrease exponentially for older data

$$\begin{aligned}\hat{y}_t &= \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots \\ &= \sum_{k=0}^{\infty} \alpha(1-\alpha)^k y_{t-k}\end{aligned}$$

- **Interpretation**
  - α: smoothing parameter ($0 < \alpha < 1$)
  - Most recent observations have the greatest influence on forecast
  - Suitable for series with no trend or seasonality
- It can be expressed in a recursive formulation:

$$\begin{aligned}\hat{y}_{t+1} &= z_t \\ z_t &= \alpha y_t + (1-\alpha) z_{t-1}\end{aligned}$$

Additive Holt-Winters Model

- Extends exponential smoothing to capture trend and seasonality (seasonal variation is roughly constant over time)
- Components of the model:
  - Level ℓ_t: the baseline value at time t
  - Trend b_t: the slope or direction of the series
  - Seasonal component s_t: repeated patterns over a fixed period
- Forecast equation:

$$\hat{y}_{t+h} = \ell_t + hb_t + s_{t+h-m(k+1)}$$

  where:
  - m: season length
  - h: forecast horizon
- Update equations:
  - Level: $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$
  - Trend: $b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}$
  - Season: $s_t = \gamma(y_t - \ell_t) + (1-\gamma)s_{t-m}$

Additive Holt-Winters Model: Use

- Intuition:
  - Smoothly updates estimates of level, trend, and season
  - Each component has its own smoothing parameter: α, β, γ
- Additive nature:
  - Adds components together: $y_t = \ell_t + b_t + s_t + \varepsilon_t$
  - Good for stable seasonal effects (e.g., monthly sales with similar variation)
- Common applications:
  - Retail sales
  - Web traffic
  - Electricity demand with fixed seasonal amplitude
  - Daily demand with seasonal shopping patterns

Vector Autoregressions (VAR)

- VAR models generalize AR models to multivariate time series
  - Each variable depends on past values of itself and others
  - E.g., for 2 variables

$$
\begin{aligned}
y_{1,t} &= c_1 + \phi_{11} y_{1,t-1} + \phi_{12} y_{2,t-1} + \varepsilon_{1,t} \\
y_{2,t} &= c_2 + \phi_{21} y_{1,t-1} + \phi_{22} y_{2,t-1} + \varepsilon_{2,t}
\end{aligned}
$$

- Intuition:
  - Capture dynamic interrelationships among multiple series
- Used for:
  - Economic indicators
  - Multichannel sensor data
- Example: modeling GDP growth and inflation jointly

Forecasting with Exogenous Variables

- ARIMAX, VARX, DeepAR architectures
    - ARIMAX (AutoRegressive Integrated Moving Average with Exogenous variables) is an extension of ARIMA that includes external variables
    - VARX (Vector AutoRegression with Exogenous variables) is used for multivariate time series data with external influences
    - DeepAR is a deep learning-based approach that can handle exogenous variables for probabilistic forecasting
- Incorporating covariates
    - Covariates are external factors that can influence the variable being forecasted
    - Common covariates include:
        - Weather conditions (e.g., temperature, precipitation)
        - Holidays and special events (e.g., Christmas, Black Friday)
        - Marketing events (e.g., promotions, advertising campaigns)
- Applications
    - Retail forecasting
        - Predicting sales based on upcoming holidays or promotional events
        - E.g., forecasting increased demand for winter clothing based on weather forecasts
    - Energy consumption
        - Estimating future energy needs based on temperature changes or public holidays
        - E.g., predicting higher electricity usage during a heatwave

Modern Approaches

- Time Series
- Classical Methods
- **Modern Approaches**
- Special techniques for time series modeling

State Space Models

- State space models describes how a system evolves over time using states and observations
- **Components**
  - State vector x_t: Hidden/internal state of the system at time t
  - Observation vector y_t: What we can measure at time t
  - The model is

$$x_{t+1} = F_t x_t + G_t u_t + w_t (\text{ State equation})$$
$$y_t = H_t x_t + v_t (\text{ Observation equation})$$

    where:
    - F_t: State transition matrix
    - G_t: Control input matrix
    - H_t: Observation matrix
    - w_t, v_t: Process and observation noise
- **Goal**
  - Infer hidden states from noisy observations
  - Predict future observations or states
- **Applications**

Frequency Domain Methods

- Analyze time series in terms of cycles and frequencies
  - E.g., Fourier Transform decomposes series into sinusoidal components
- Intuition:
  - Understand repeating patterns that may not be obvious in time domain
- Useful for:
  - Identifying dominant periodicities
  - Filtering noise
- Applications:
  - Seismology
  - Climate cycles
- Example: detect yearly cycle in temperature data

Machine Learning for Time Series

- Use supervised learning to predict future values
- Feature engineering, e.g.,
  - Lags
  - Rolling values
  - Fourier terms
- Common algorithms:
  - Decision trees
  - Random forests
  - Gradient boosting (e.g., XGBoost)
- Pros
  - Handle nonlinearity and complex interactions
- Cons
  - Often requires careful cross-validation due to temporal structure
- Example: predicting electricity consumption using lagged features

Deep Learning for Time Series

- Specialized neural networks for sequential data
  - Recurrent Neural Networks (RNNs):
    - Capture dependencies across time steps
  - Long Short-Term Memory networks (LSTMs):
    - Solve vanishing gradient problem
    - Retain long-term dependencies
  - Temporal Convolutional Networks (TCNs):
    - Use causal convolutions for sequence modeling
- Pros:
  - Handle complex, nonlinear dynamics
- Cons:
  - Require large datasets and careful tuning
- Example: predicting stock price movements using past sequences

Bayesian Time Series Models

- Incorporate prior beliefs and uncertainty into time series modeling
  - **Bayesian ARIMA**:
    - Extends classical ARIMA with prior distributions on parameters
    - Captures uncertainty in predictions and parameter estimates
  - **Bayesian State Space Models**:
    - General framework for modeling hidden processes over time
    - Includes Kalman Filters and extensions with Bayesian inference
- Solve with MCMC (Markov Chain Monte Carlo):
  - Key method for posterior inference in Bayesian models
  - Generates samples from complex posterior distributions
  - Probabilistic programming tools, e.g., PyMC
- Pros
  - Enables rich, interpretable models with quantified uncertainty
  - Suitable for forecasting, anomaly detection, and causal inference
- Cons
  - Computationally expensive

Special techniques for time series modeling

- Time Series
- Classical Methods
- Modern Approaches
- **Special techniques for time series modeling**

Time Series Machine Learning: Problems

- **Time Dependence**
  - Observations are not independent and identically distributed (i.i.d.)
  - Temporal ordering affects model assumptions and evaluation
  - Past values can strongly influence future ones (autocorrelation)
- **Non-Stationarity**
  - Data distributions change over time (mean, variance, etc.)
  - Models trained on past data may perform poorly on future data
- **Data Leakage**
  - Using future information in training can lead to over-optimistic results
  - Requires careful data splitting (e.g., no random shuffling)
- **Seasonality and Trends**
  - Regular patterns (e.g., daily, yearly) complicate modeling
  - Need to be explicitly handled or removed
- **Missing or Irregular Data**
  - Time series often have missing timestamps or irregular intervals
- **Evaluation Challenges**
  - Time-aware cross-validation needed (e.g., rolling windows)
- **Feature Engineering Complexity**
  - Lagged variables, rolling statistics, and domain-specific features needed
  - Makes pipeline design more involved compared to i.i.d. data

Cross-Validation for Time Series

- Standard cross-validation is not suitable due to time dependency
- **Walk-forward validation** is a time-aware cross-validation method for time series data
  - Mimics real-world usage: train on past, predict future
  - Choose a fixed-size training window
  - Move the training and testing windows forward in time step-by-step
  - Train and evaluate model performance at each step
- **Example**:
  - Step 1: Train on $[t_1, t_{100}]$, test on t_{101}
  - Step 2: Train on $[t_2, t_{101}]$, test on t_{102}
  - Continue moving forward through the time series
- **Pros**:
  - Preserves time order
  - Provides multiple performance estimates across time
- **Cons**
  - Machine learning becomes even more complicated
  - Conflates generalization error and non-stationarity

Anomaly Detection in Time Series

- Identify unusual patterns not consistent with past behavior
    - Important to account for seasonality and trend
    - Unsupervised methods are often necessary
- Common methods:
    - Statistical thresholds
        - Gaussian modeling and consider anomaly anything $> 3\sigma$ from mean
    - Machine learning, e.g.,
        - Trees
        - Autoencoders
- Applications:
    - Finance: fraud detection, unusual trading activity
    - Cybersecurity: intrusion detection, system failures
    - Example: detecting a sudden drop in website traffic

Probabilistic Forecasting

- Predict full distribution of future values, not just a single number
  - Useful when risk-sensitive decisions depend on forecast range
  - Helps express uncertainty explicitly
  - E.g., forecasting a 90% prediction interval for electricity demand
  - E.g., finance, weather forecasting, and supply chain management
- Common methods:
  - Quantile regression
    - Predict specific quantiles (e.g., 10%, 50%, 90%)
    - E.g., estimating the 10th percentile of future stock prices to assess downside risk
  - Bayesian models
    - Incorporate prior knowledge and update beliefs with new data
    - E.g., using Bayesian models to update demand forecasts as new sales data becomes available
  - Other methods:
    - Bootstrapping techniques to generate prediction intervals
    - Ensemble methods that combine multiple models to improve forecast accuracy

Granger Causality

- A statistical hypothesis test for determining causality in time series
  - Variable X *Granger-causes* Y if past values of X contain information that helps predict Y beyond the past values of Y alone
  - E.g., fit two Vector Autoregressive (VAR) models:
    - Restricted model: only past Y
    - Unrestricted model: past Y and past X
    - Test null hypothesis: X does not Granger-cause Y
- Cons:
  - Predictive, not necessarily true causal influence
  - Based on temporal precedence and improvement in forecast
  - Need to measure out-of-sample
  - Need to account for model complexity
  - Sensitive to model specification (e.g., lag choice and non-linearity)
- Applications:
  - Macroeconomics: does money supply affect inflation?
  - Finance: does trading volume predict stock returns?
  - Neuroscience: do neural signals in one region predict another?

Change Point Detection in Time Series

- Identify times when the statistical properties of a time series change, e.g.,
  - Mean shift
  - Variance change
  - Trend change
  - Distribution change
- **Methods**
  - Offline detection: analyze full data set
    - Useful when historical data is available
    - E.g., analyzing stock prices over the past year to detect shifts
  - Online detection: identify change in real-time
    - Crucial for applications requiring immediate response
    - E.g., monitoring network traffic for anomalies
- **Approaches**
  - Model-based methods rely on assumptions about data distribution
    - E.g., ARIMA models for time series forecasting
  - Bayesian approaches:
    - Model changes with probabilistic assumptions and priors
- **Applications**
  - Financial market regime shifts
    - Detecting bull and bear markets
  - Environmental change
    - Monitoring climate data for significant shifts

Markov-Switching Models

- Time series models that allow regime changes over time
  - Data is generated by multiple underlying regimes (states)
  - Regimes switch according to a Markov process
  - Each regime
    - Has its own parameters (e.g., mean, variance, AR coefficients)
    - Can represent different behaviors (e.g., growth vs. recession)
- **Markov Property**:
  - Probability of switching depends only on the current state
  - Defined by a state transition matrix
- **Applications**:
  - Economics: modeling business cycles
  - Finance: volatility modeling (e.g., bull vs. bear markets)
- Pros:
  - Captures nonlinear dynamics and structural breaks in time series