Deep Temporal Models

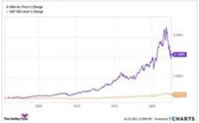
Sourangshu Bhattacharya
Department of Computer Science and Engg.
IIT Kharagpur
https://cse.iitkgp.ac.in/~sourangshu/

Time Series Analysis

Time Series Data is Ubiquitous

- A wide range of time series data
  - AIOps
  - IoT
  - Business data, e.g., sales volume, stock price
  - Many others

stocks

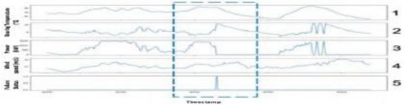


sales

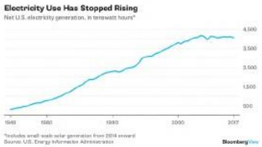
Light vehicle retail sales in the United States from 1976 to 2021 (in 1,000 units)

goods consumption

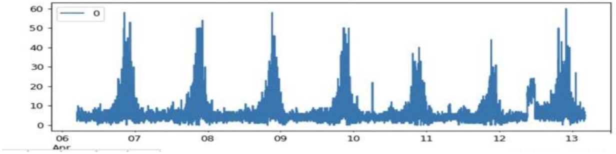
sensor



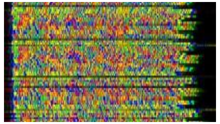
power demand



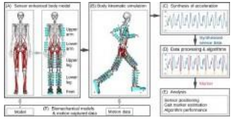
Cloud service monitoring



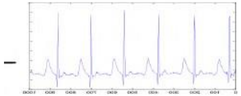
DNA sequence



motion detect

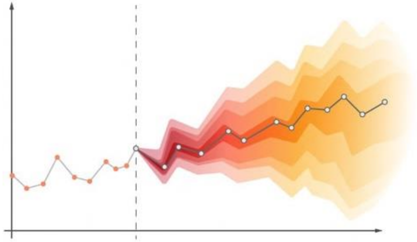


ECG

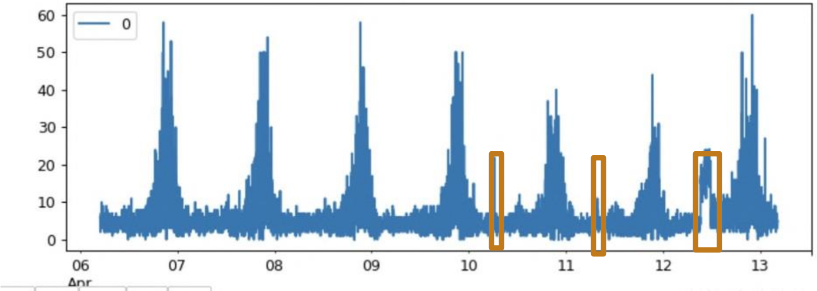


Typical Applications of Time Series

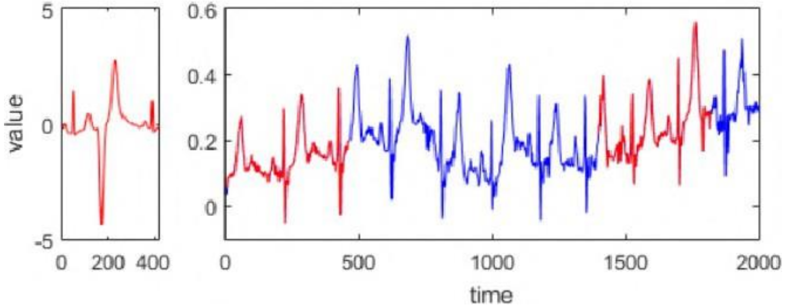
Time Series Forecasting



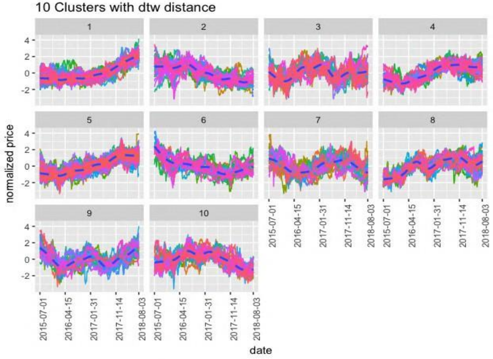
Time Series Anomaly Detection



Time Series Search/Query

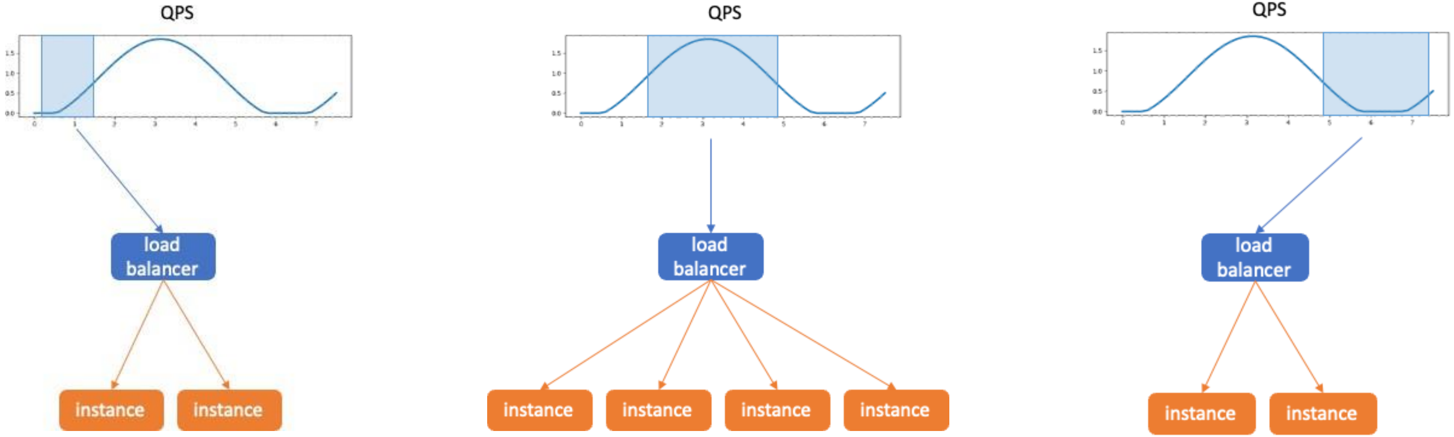


Time Series Classification/Clustering



Forecasting Use Case: AutoScaling

- Autoscaling in cloud computing is an effective method to improve the usage of computing resources
  - It automatically allocates resources for cloud-based applications while maintaining SLA (service level agreement)
  - Horizontal scaling (add/delete instances or VMs) vs vertical scaling (up/downgrade CPU, RAM, network, etc.)
  - Time series forecasting and decision-making on resources



Introduction

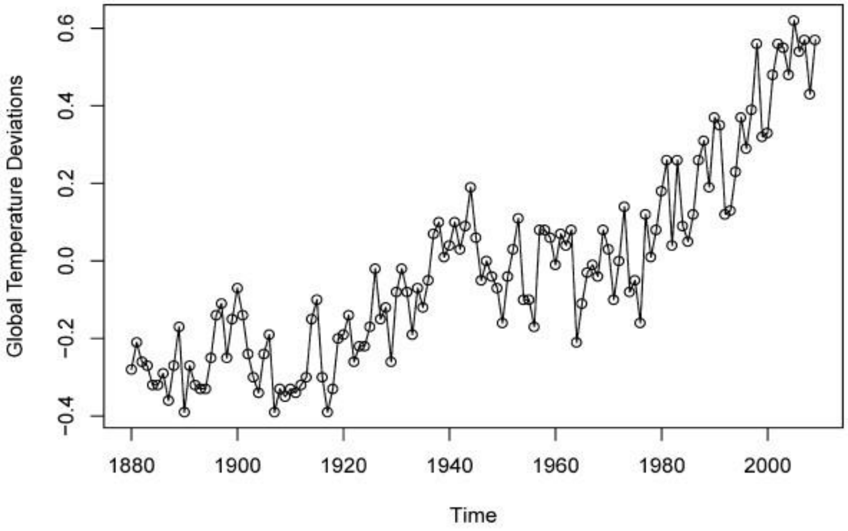
Plotting a time series is an important early step in its analysis In general, a plot can reveal:

- *Trend*: upward or downward pattern that might be extrapolated into the future
- *Periodicity*: Repetition of behavior in a regular pattern
- *Seasonality*: Periodic behavior with a known period (hourly, monthly, every 2 months...)
- *Heteroskedasticity*: changing variance
- *Dependence*: positive (successive observations are similar) or negative (successive observations are dissimilar)
- Missing data, outliers, breaks...

Example Time Series

Example 1: Global Warming

The data are the global mean land-ocean temperature index from 1880 to 2009. We note an *apparent upward trend* in the series during the latter part of the 20th century that has been used as an argument for the global warming hypothesis (whether the overall trend is natural or whether it is caused by some human-induced interface)

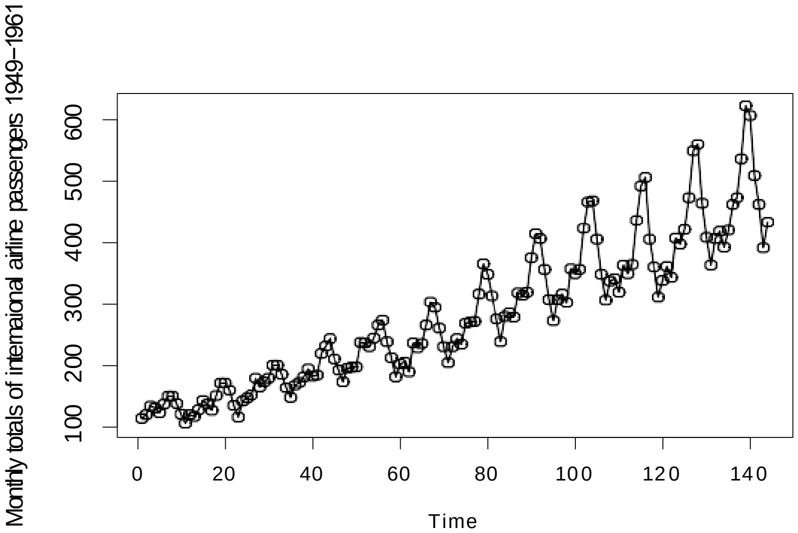


Example Time Series

Example 4: Airline passengers from 1949-1961

Trend? Seasonality? Heteroskedasticity? …

Upward trend, seasonality on a 12 month interval, increasing variability

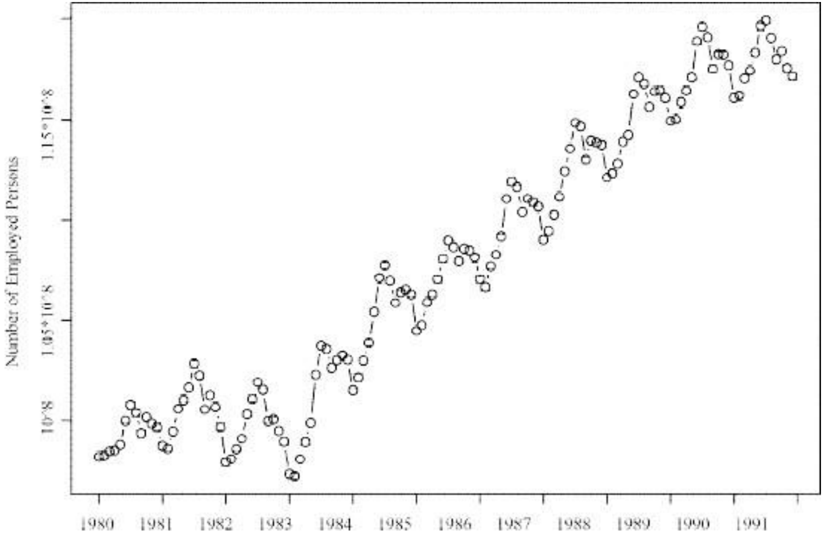


Example Time Series

Example 5: Monthly Employed persons from 1980-1991

Trend? Seasonality? Heteroskedasticity? …

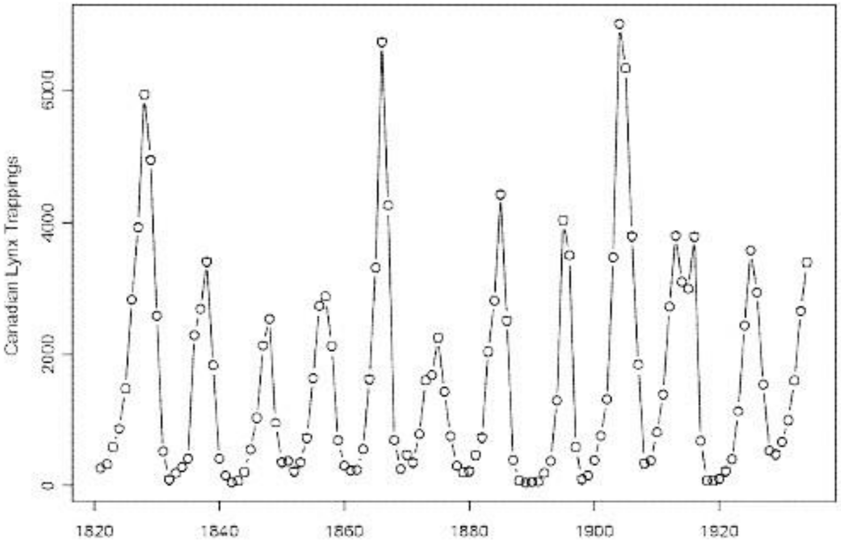
Upward trend, seasonality with a structural break



Example Time Series

Example 7: Annual number of Canadian Lynx trapped near McKenzie River

Trend? Seasonality? Heteroskedasticity? breaks?... **no trend, no clear seasonality as it does correspond to a known period, periodicity**



Objectives of Time Series Analysis

What do we hope to achieve with time series analysis?

- Provide a model of the data (testing of scientific hypothesis, etc.)

- Predict future values (very common goal of analysis)

- Produce a compact description of the data (a good model can be used for "data compression")

Modeling Time Series

We take the approach that the data is a realization of random variable.

However, many statistical tools are based on assuming any R.V. are IID.

In Times Series:

R.V. are usually not independent (affected by trend and seasonality)

Variance may change significantly

R.V. are usually not identically distributed

The first goal in time series modeling **is to reduce the analysis needed to a simpler case**: Eliminate Trend, Seasonality, and heteroskedasticity then we model the remainder as dependent but Identically distributed.

Probabilistic Model: Stochastic Process

A complete probabilistic model/description of a time series X_t observed as a collection of n random variables at times $t_1, t_2, . . . , t_n$ for any positive integer n is provided by the joint probability distribution,

$$F(C_1, C_2, ..., C_n) = P(X_1 \leq C_1, ..., X_n \leq C_n)$$

This is generally difficult to write, unless the case the variables are jointly normal.

Thus, we look for other statistical tools => quantifying dependencies

Properties of Time Series Model

A time series model is a Discrete Time Stochastic Process.

A time series model for the observed data X_t

The mean function $\mu_X = E(X_t)$

The Covariance function
$\gamma_X(r, s) = E((X_r - \mu_X(r))(X_s - \mu_X(s)))$ for all integers r and s

The focus will be to determine the mean function and the Covariance function to define the time series model.

Some zero-Mean Models

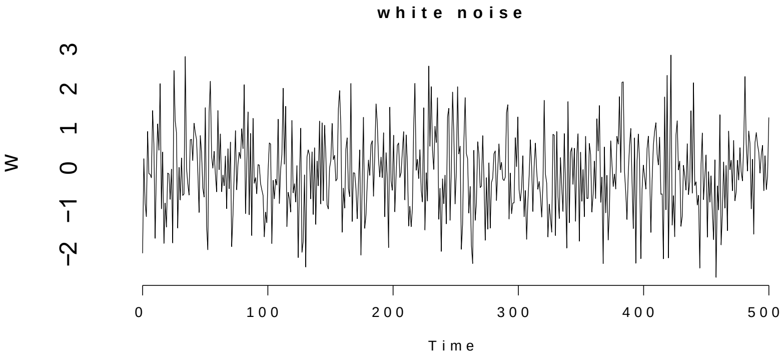
iid Noise

The simplest model for a times series: no trend or seasonal component and in which the observations are IID with zero mean.

We can write, for any integer n and real numbers $x_1, x_2, \ldots, x_n$,

$$P(X_1 \leq x_1, \ldots, X_n \leq x_n) = P(X_1 \leq x_1) \ldots P(X_n \leq x_n)$$

It plays an important role as a building block for more complicated time series models



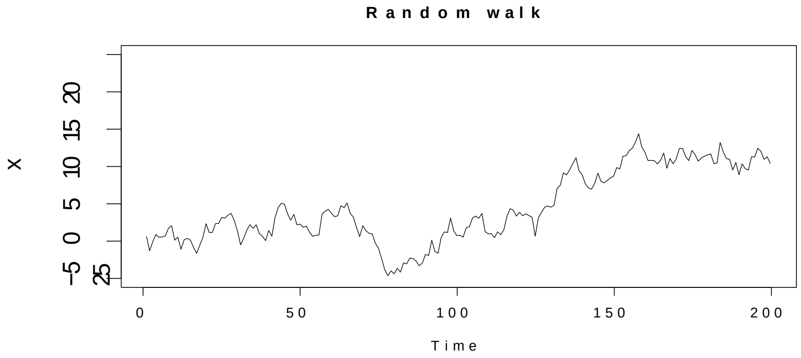
Some zero-Mean Models

Random Walk

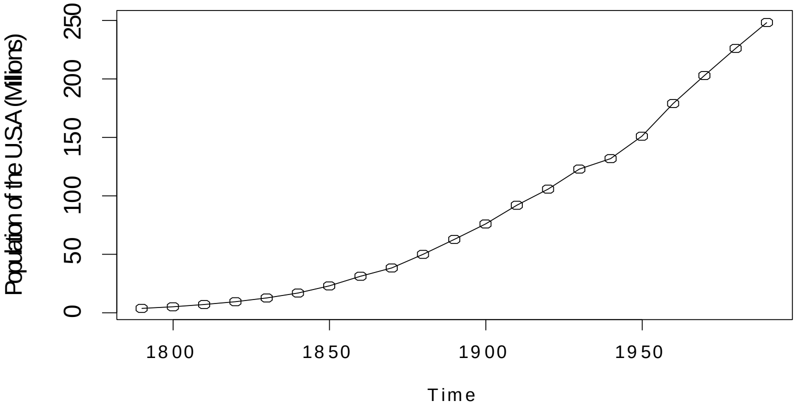
The random walk $\{S_t\}$, $t = 0, 1, 2, ...$ is obtained by cumulatively summing iid random variables, $S_0 = 0$

$$S_t = X_1 + X_2 + \cdots + X_t, \qquad t = 1, 2, ...$$

where X_t is iid noise. It plays an important role as a building block for more complicated time series models



Models with Trend

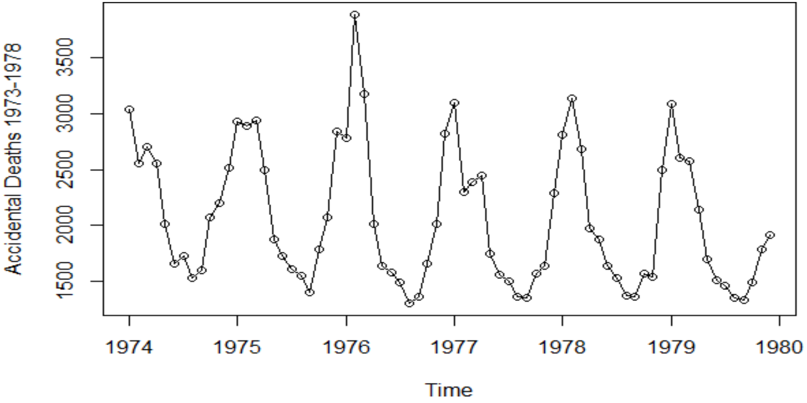


In this case a zero-mean model for the data is clearly inappropriate. The graph suggests trying a model of the form:

$$X_t = m_t + Y_t$$

where m_t is a function known as the trend component and Y_t has a zero mean. Estimating m_t?

Models with Seasonality



In this case a zero-mean model for the data is clearly inappropriate. The graph suggests trying a model of the form:

$$X_t = S_t + Y_t$$

where S_t is a function known as the season component and Y_t has a zero mean. Estimating S_t?

Time series Modeling

Plot the series => examine the main characteristics (trend, seasonality, ...)

Remove the trend and seasonal components to get stationary residuals/models

Choose a model to fit the residuals using sample statistics (sample autocorrelation function)

Forecasting will be given by forecasting the residuals to arrive at forecasts of the original series X_t

Stationary and Autocorrelation function

Definitions

X_t is **strictly** stationary if $\{X_1, \ldots X_n\}$ and $\{X_{1+h}, \ldots X_{n+h}\}$ have the same joint distributions for all integers h and $n > 0$.

X_t is **weakly** stationary if

$\mu_X(t)$ is independent of t.

$\gamma_X(t+h, t)$ is independent of t for each h.

Let X_t be a stationary time series. **The autocovariance function** (ACVF) of X_t at lag h is

$$\gamma_X(h) = Cov(X_{t+h}, X_t)$$

The **autocorrelation function** (ACF) of X_t at lag h is

$$\rho_X(h) = \frac{\gamma_X(h)}{\gamma_X(0)} = Cor(X_{t+h}, X_t)$$

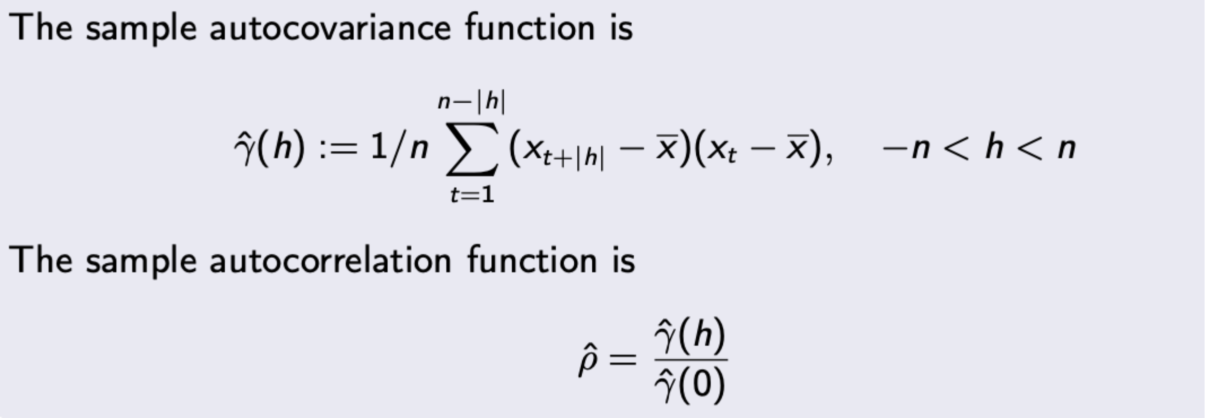
The Sample Autocorrelation function

In practical problems, we do not start with a model, but with observed data $(x_1, x_2, \ldots, x_n)$. **To assess the degree of dependence** in the data and to **select a model** for the data, one of the important tools we use is the sample autocorrelation function (Sample ACF).

Definition

Let $x_1, x_2, \ldots, x_n$ be observations of a time series. The sample mean of $x_1, x_2, \ldots, x_n$ is

$$\bar{X} = \frac{1}{n}\sum_{t=1}^{n} x_t$$



The sample autocovariance function is

$$\hat{\gamma}(h) := 1/n \sum_{t=1}^{n-|h|} (x_{t+|h|} - \bar{x})(x_t - \bar{x}), \quad -n < h < n$$

The sample autocorrelation function is

$$\hat{\rho} = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}$$

The Sample Autocorrelation function

Remarks

The sample autocorrelation function (ACF) can be computed for any data set and is not restricted to observations from a stationary time series.

For data containing a Trend, $|\hat{\rho}(h)|$ will display slow decay as h increases.

For data containing a substantial deterministic periodic component, $|\hat{\rho}(h)|$ will exhibit similar behavior with the same periodicity.

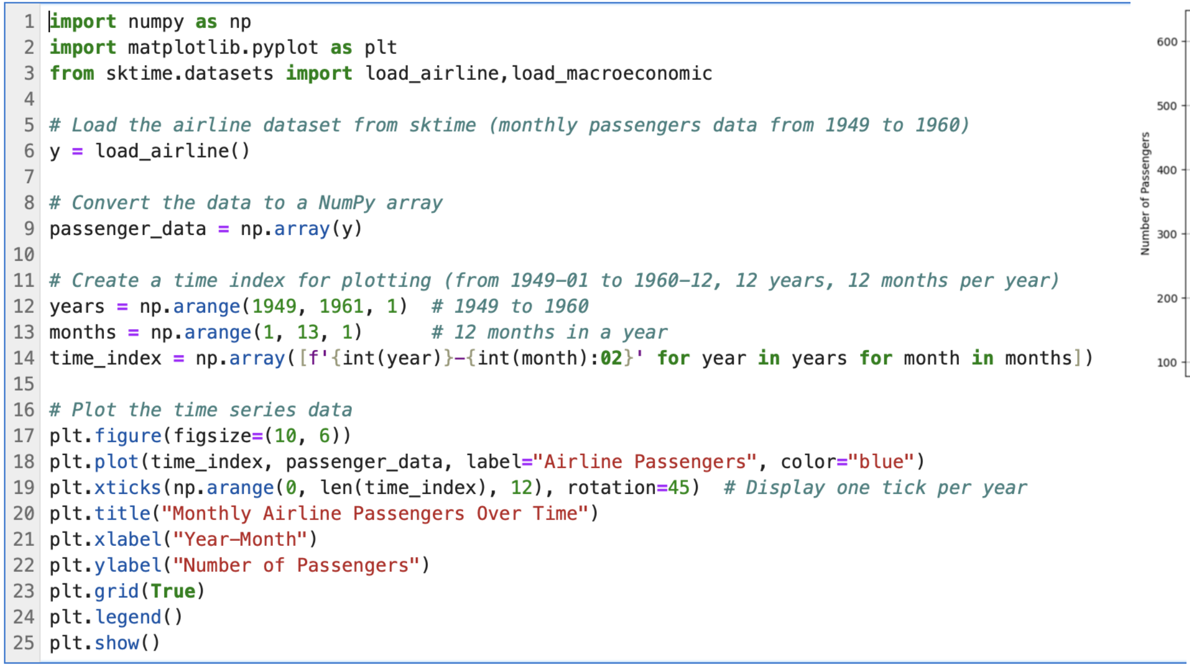
The Sample Autocorrelation function

Remarks

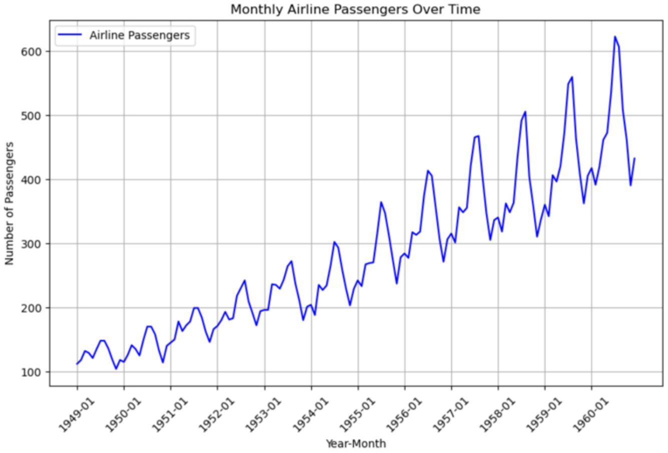
We may recognize the sample autocorrelation function of many time series:

- White Noise => Zero Trend => Slow decay
- Periodic => Periodic
- Moving Average (q) => Zero for $|h| > q$
- AutoRegression (p) => Decay to zero exponentially

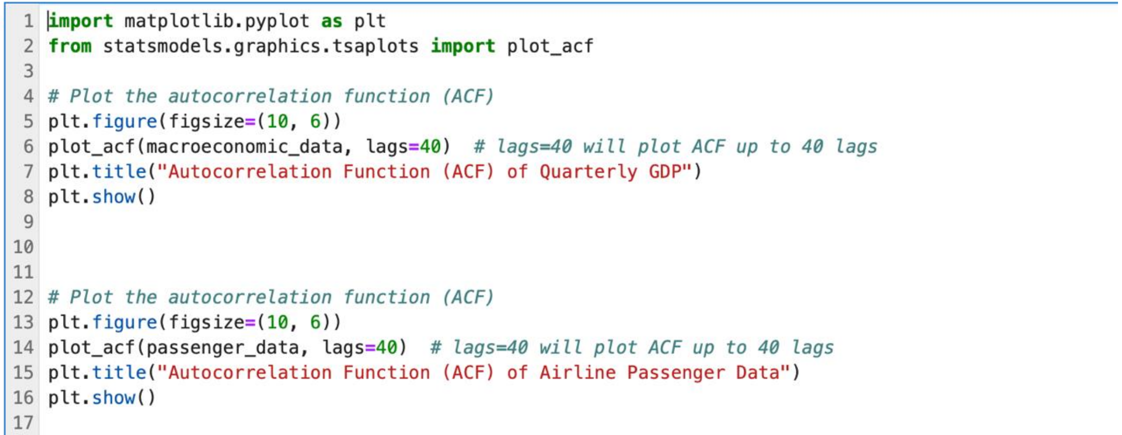
The Airlines Dataset



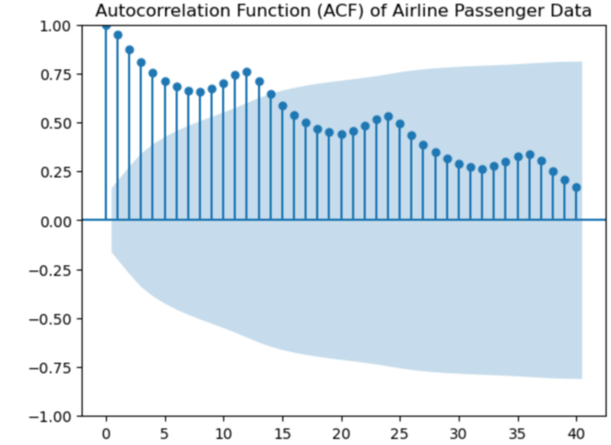
```python
import numpy as np
import matplotlib.pyplot as plt
from sktime.datasets import load_airline,load_macroeconomic

# Load the airline dataset from sktime (monthly passengers data from 1949 to 1960)
y = load_airline()

# Convert the data to a NumPy array
passenger_data = np.array(y)

# Create a time index for plotting (from 1949-01 to 1960-12, 12 years, 12 months per year)
years = np.arange(1949, 1961, 1)  # 1949 to 1960
months = np.arange(1, 13, 1)      # 12 months in a year
time_index = np.array([f'{int(year)}-{int(month):02}' for year in years for month in months])

# Plot the time series data
plt.figure(figsize=(10, 6))
plt.plot(time_index, passenger_data, label="Airline Passengers", color="blue")
plt.xticks(np.arange(0, len(time_index), 12), rotation=45)  # Display one tick per year
plt.title("Monthly Airline Passengers Over Time")
plt.xlabel("Year-Month")
plt.ylabel("Number of Passengers")
plt.grid(True)
plt.legend()
plt.show()
```

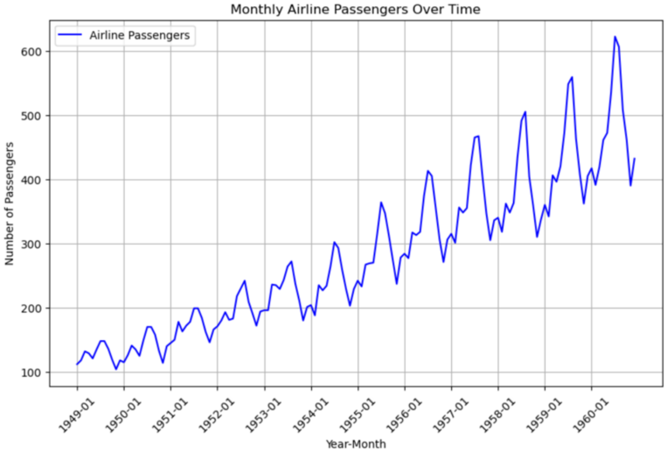


The Sample Autocorrelation function



```python
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf

# Plot the autocorrelation function (ACF)
plt.figure(figsize=(10, 6))
plot_acf(macroeconomic_data, lags=40)  # lags=40 will plot ACF up to 40 lags
plt.title("Autocorrelation Function (ACF) of Quarterly GDP")
plt.show()



# Plot the autocorrelation function (ACF)
plt.figure(figsize=(10, 6))
plot_acf(passenger_data, lags=40)  # lags=40 will plot ACF up to 40 lags
plt.title("Autocorrelation Function (ACF) of Airline Passenger Data")
plt.show()


```

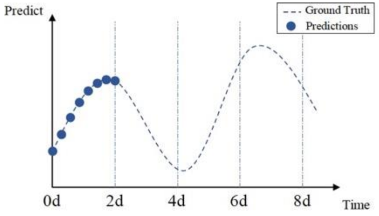




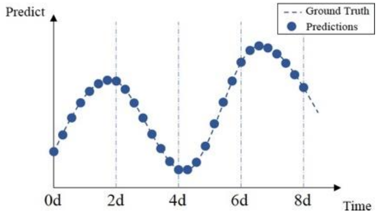
Time Series Forecasting

Forecasting: Background

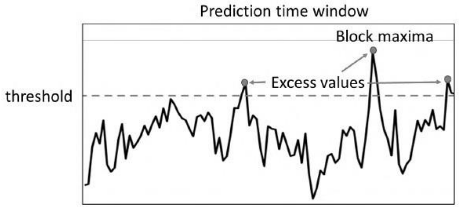
- Different forecasting types
  - **Short-term** forecasting: predict the near future
  - **Long-term** forecasting: predict the future with an extended period
  - **Extreme value** forecasting: predict the extreme values
  - **Point or Probabilistic** forecasting: predict point value or interval/probability distribution
- Challenges:
  - Accuracy, robustness
- Models:
  - Traditional: Statistical (ARIMA, ETS, Prophet)
  - Ensemble: Tree, MLP
  - Deep Models: CNN, RNN, Transformers



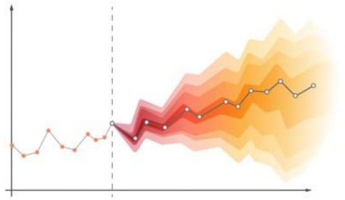
Short term forecasting



Long term forecasting



Extreme value forecasting



Probabilistic forecasting

ARIMA Models: General framework

An ARIMA model is a numerical expression indicating how the observations of a target **variable are statistically correlated with past observations of the same variable**

- ARIMA models are, in theory, the most general class of models for forecasting a time series which can be "**stationarized**" by transformations such as differencing and lagging

- The easiest way to think of ARIMA models is as fine-tuned versions of random-walk models: the fine-tuning consists of adding lags of the differenced series and/or lags of the forecast errors to the prediction equation, as needed to remove any remains of autocorrelation from the forecast errors

In an ARIMA model, in its most complete formulation, are considered:

- An **Autoregressive (AR)** component, seasonal and not
- A **Moving Average (MA)** component, seasonal and not
- The order of **Integration (I)** of the series

That's why we call it ARIMA (Autoregressive Integrated Moving Average)

ARIMA Models: General framework

The most common notation used for ARIMA models is:

$$\boldsymbol{ARIMA(p, d, q)}$$

where:

- **p** is the number of autoregressive terms
- **d** is the number of non-seasonal differences
- **q** is the number of lagged forecast errors in the equation

- **In the next slides we will explain each single component of ARIMA models!**

ARIMA Models: Autoregressive part (AR)

In a **multiple regression model**, we predict the target variable Y using a linear combination of independent variables (predictors)

In an **autoregression model**, we forecast the variable of interest using a linear combination of past values of the variable itself

The term autoregression indicates that it is a regression of the variable against itself

- An **Autoregressive model of order $\boldsymbol{p}$**, denoted $AR(p)$ model, can be written as

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

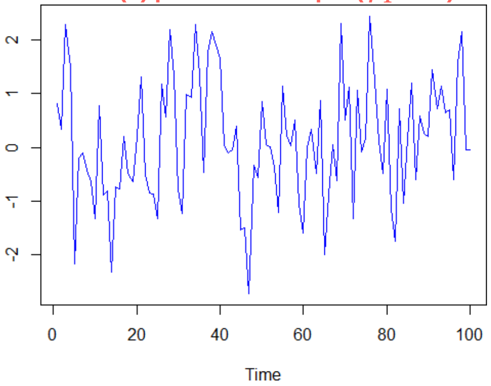
Where:

- y_t = dependent variable
- $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$= independent variables (i.e. lagged values of y_t as predictors)
- $\phi_1, \phi_2, \ldots, \phi_p$ = regression coefficients
- ε_t= error term (must be white noise)

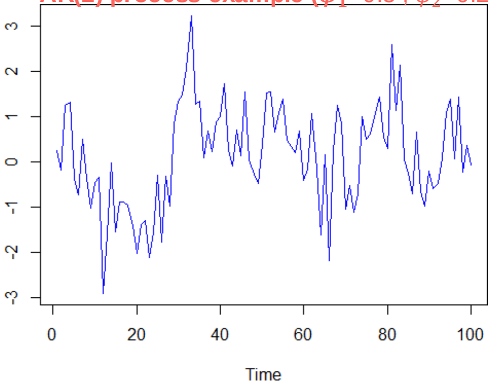
ARIMA Models: Autoregressive part (AR)

Autoregressive simulated process examples:

AR(1) process example (ϕ_1=0.5)



AR(2) process example (ϕ_1=0.5 , ϕ_2=0.2)



Consider that, in case of **AR(1)** model:

- When $\phi_1 = 0$, y_t is a white noise
- When $\phi_1 = 1$ and $c = 0$, y_t is a random walk
- In order to have a stationary series the following condition must be true: $-1 < \phi_1 < 1$

ARIMA Models: Moving Average part (MA)

Rather than use past values of the forecast variable in a regression, a Moving Average model uses **past forecast errors** in a regression-like model

In general, a moving average process of order q, MA (q), is defined as:

$$y_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q}$$

The lagged values of ε_t are not actually observed, so it is not a standard regression.

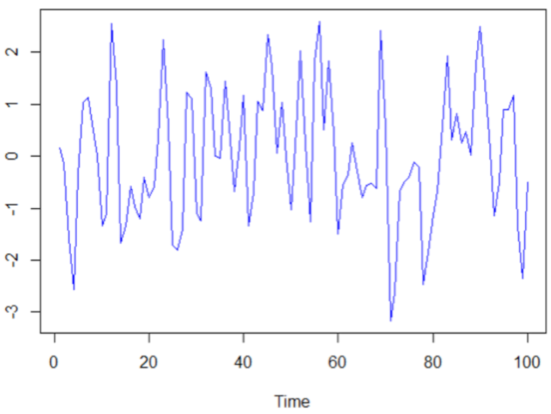
Moving average models should not be confused with moving average smoothing (the process used in classical decomposition in order to obtain the trend component)

A moving average model is used for forecasting future values while moving average smoothing is used for estimating the trend-cycle of past values

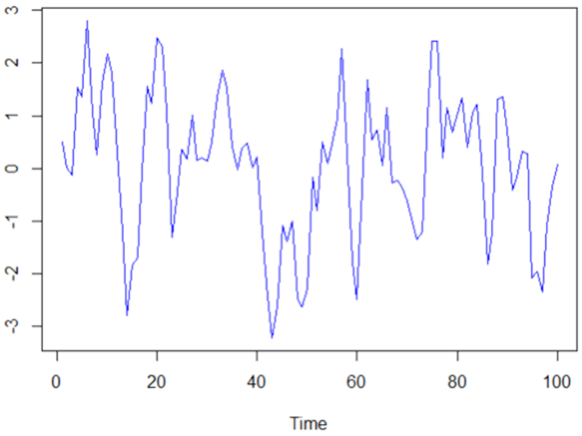
ARIMA Models: Moving Average part (MA)

Moving Average simulated process examples:

MA(1) process example (θ_1=0.7)



MA(2) process example (θ_1=0.8 , θ_2=0.5)



- Looking just the time plot it’s hard to distinguish between an AR process and a MA process!

ARIMA Models: ARMA and ARIMA

If we combine autoregression and a moving average model, we obtain an **ARMA(p,q)** model:

$$y_t = c + \underbrace{\phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p}}_{\text{Autoregressive component of order p}} + \underbrace{\theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}}_{\text{Moving Average component of order q}} + \varepsilon_t$$

To use an ARMA model, the series must be **STATIONARY**!

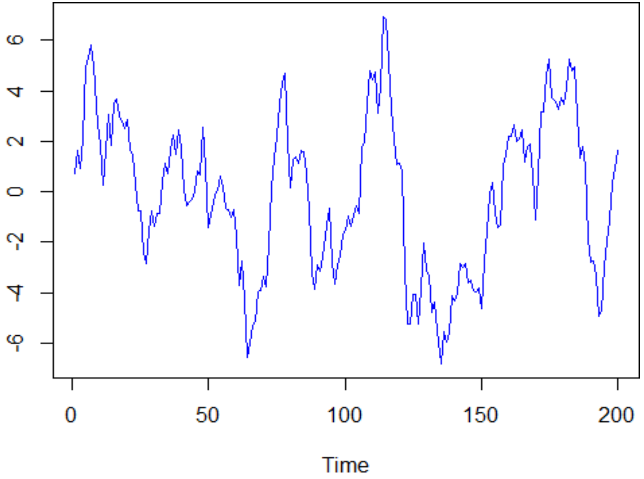
- If the series is NOT stationary, before estimating and ARMA model, we need to apply one or more differences in order to make the series stationary: this is the integration process, called ***I(d)***, where d= number of differences needed to get stationarity

- If we model *the integrated* series using an ARMA model, we get an **ARIMA (p,d,q)** model where p=order of the autoregressive part; d=order of integration; q= order of the moving average part

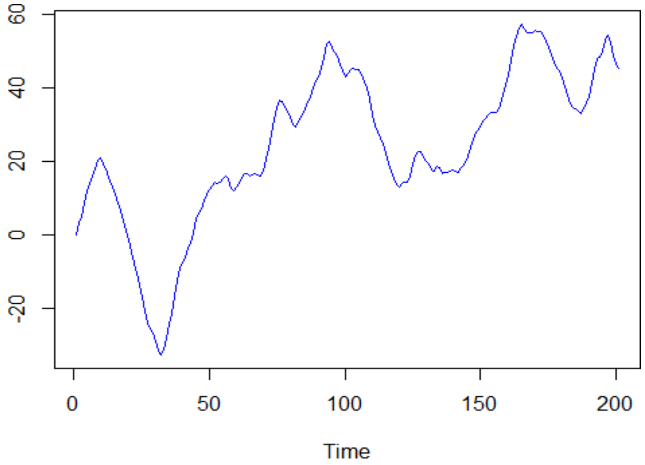
ARIMA Models: ARMA and ARIMA

ARIMA simulated process examples

ARMA(2,1) process example, equal to ARIMA(2,0,1) (ϕ_1=0.5, ϕ_2=0.4, θ_1=0.8)



ARIMA(2,1,1) process example (ϕ_1=0.5, ϕ_2=0.4, θ_1=0.8)



ARIMA Models: Model identification

General rules for model indentification based on ACF and PACF plots:

The data may follow an $\boldsymbol{ARIMA(p, d, 0)}$ model if the ACF plots of the differenced data show the following patterns:

- the ACF is exponentially decaying or sinusoidal

The data may follow an $\boldsymbol{ARIMA(0, d, q)}$ model if the ACF plots of the differenced data show the following patterns:

- the PACF is exponentially decaying or sinusoidal

□ For a general $\boldsymbol{ARIMA(p, d, q)}$ model (with both **p** and **q > 1**) both ACF and PACF plots show exponential or sinusoidal decay and it’s more difficult to understand the structure of the model

ARIMA Models: Seasonal ARIMA

A seasonal ARIMA model is formed by including **additional seasonal terms in the ARIMA models** we have seen so far

$$ARIMA\underbrace{(p,d,q)}_{\substack{\uparrow\\ \begin{pmatrix}\text{Non-seasonal part}\\ \text{of the model}\end{pmatrix}}}\ \underbrace{(P,D,Q)}_{\substack{\uparrow\\ \begin{pmatrix}\text{Seasonal part}\\ \text{of the model}\end{pmatrix}}}s$$

where s = number of periods per season (i.e. the frequency of seasonal cycle)

We use uppercase notation for the seasonal parts of the model, and lowercase notation for the non-seasonal parts of the model

- As usual, d / D are the number of differences/seasonal differences necessary to make the series stationary

AI4ICPS IIT Kharagpur

ARIMA Models: estimation and AIC

Parameters estimation

In order to estimate an ARIMA model, normally it’s used the **Maximum Likelihood Estimation (MLE)**

This technique **finds the values of the parameters which maximize the probability of obtaining the data that we have observed** □ For *given values* of $(\boldsymbol{p},\ \boldsymbol{d},\ \boldsymbol{q})$ $(\boldsymbol{P},\ \boldsymbol{D},\ \boldsymbol{Q})$ (i.e. model order) the algorithm will try to **maximize the log likelihood** when finding parameter estimates

ARIMA model order

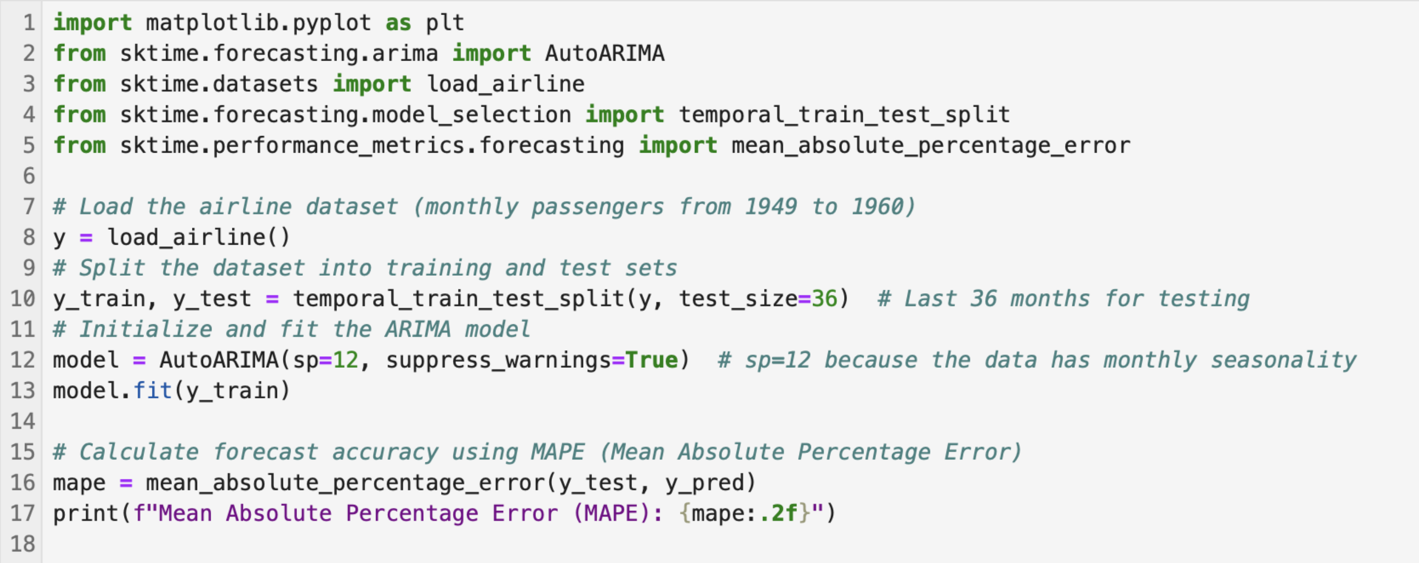
A commonly used criteria to compare different ARIMA models (i.e. with different values for $(\boldsymbol{p},\boldsymbol{q})$ $(\boldsymbol{P},\boldsymbol{Q})$ but fixed $\boldsymbol{d}$ $,\boldsymbol{D}$) and to determine the optimal ARIMA order, is the **Akaike Information Criterion (AIC)**

$$\mathbf{AIC} = -2\log\left(Likelihood\right) + 2(p)$$

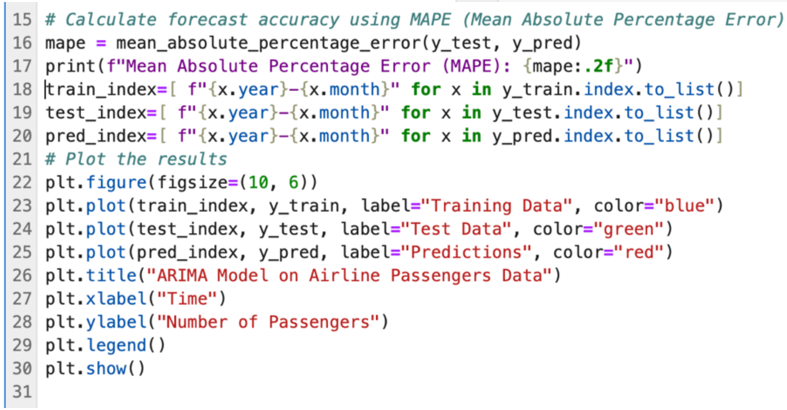
- where *p* is the number of estimated parameters in the model
- AIC is a goodness of fit measure
- **The best ARIMA model is that with the lower AIC** □ most of automatic model selection method (e.g *auto.arima* in R) uses the AIC for determining the optimal ARIMA model order

AI4ICPS IIT Kharagpur

ARIMA Models: Hands on

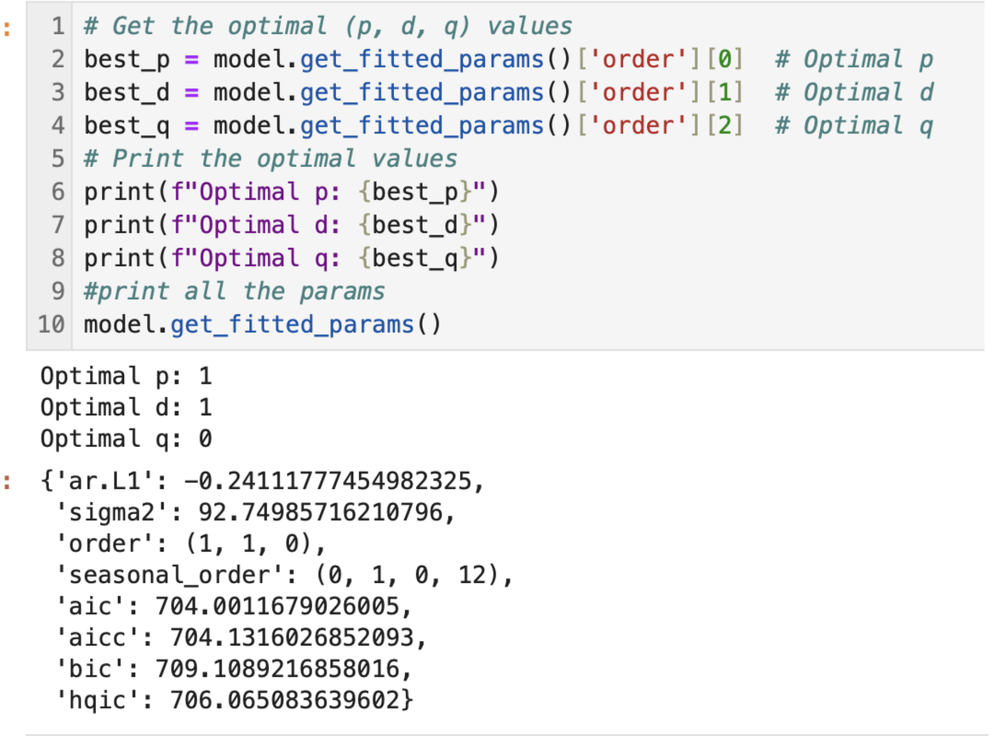


```python
import matplotlib.pyplot as plt
from sktime.forecasting.arima import AutoARIMA
from sktime.datasets import load_airline
from sktime.forecasting.model_selection import temporal_train_test_split
from sktime.performance_metrics.forecasting import mean_absolute_percentage_error

# Load the airline dataset (monthly passengers from 1949 to 1960)
y = load_airline()
# Split the dataset into training and test sets
y_train, y_test = temporal_train_test_split(y, test_size=36)  # Last 36 months for testing
# Initialize and fit the ARIMA model
model = AutoARIMA(sp=12, suppress_warnings=True)  # sp=12 because the data has monthly seasonality
model.fit(y_train)

# Calculate forecast accuracy using MAPE (Mean Absolute Percentage Error)
mape = mean_absolute_percentage_error(y_test, y_pred)
print(f"Mean Absolute Percentage Error (MAPE): {mape:.2f}")

```



```python
# Calculate forecast accuracy using MAPE (Mean Absolute Percentage Error)
mape = mean_absolute_percentage_error(y_test, y_pred)
print(f"Mean Absolute Percentage Error (MAPE): {mape:.2f}")
train_index=[ f"{x.year}-{x.month}" for x in y_train.index.to_list()]
test_index=[ f"{x.year}-{x.month}" for x in y_test.index.to_list()]
pred_index=[ f"{x.year}-{x.month}" for x in y_pred.index.to_list()]
# Plot the results
plt.figure(figsize=(10, 6))
plt.plot(train_index, y_train, label="Training Data", color="blue")
plt.plot(test_index, y_test, label="Test Data", color="green")
plt.plot(pred_index, y_pred, label="Predictions", color="red")
plt.title("ARIMA Model on Airline Passengers Data")
plt.xlabel("Time")
plt.ylabel("Number of Passengers")
plt.legend()
plt.show()

```

ARIMA Models: Hands on

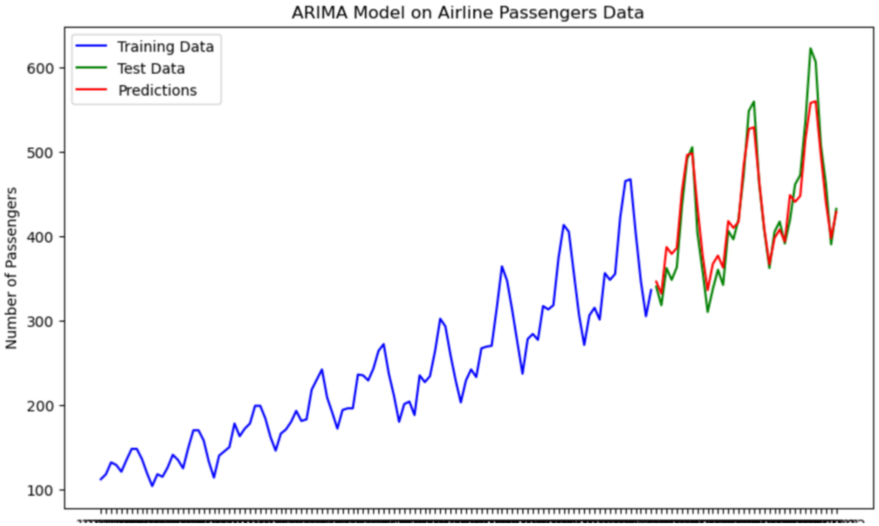


```python
# Get the optimal (p, d, q) values
best_p = model.get_fitted_params()['order'][0]  # Optimal p
best_d = model.get_fitted_params()['order'][1]  # Optimal d
best_q = model.get_fitted_params()['order'][2]  # Optimal q
# Print the optimal values
print(f"Optimal p: {best_p}")
print(f"Optimal d: {best_d}")
print(f"Optimal q: {best_q}")
#print all the params
model.get_fitted_params()
```

```
Optimal p: 1
Optimal d: 1
Optimal q: 0
```

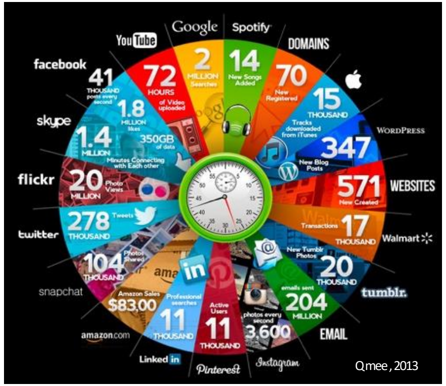
```
{'ar.L1': -0.24111777454982325,
 'sigma2': 92.74985716210796,
 'order': (1, 1, 0),
 'seasonal_order': (0, 1, 0, 12),
 'aic': 704.0011679026005,
 'aicc': 704.1316026852093,
 'bic': 709.1089216858016,
 'hqic': 706.065083639602}
```

```
Mean Absolute Percentage Error (MAPE): 0.04
```



Temporal Point Process

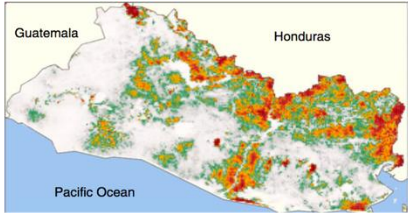
Many discrete *events* in continuous time



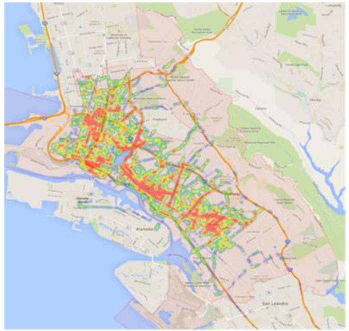
Qmee , 2013

Online actions

Financial trading



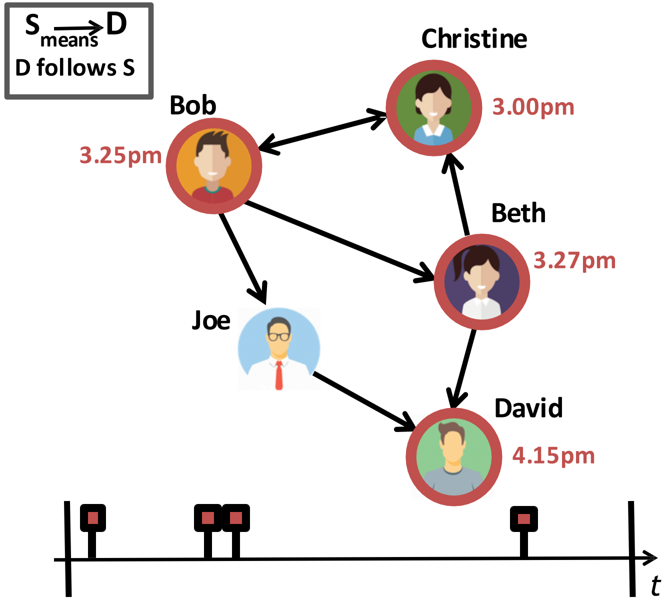
Disease dynamics

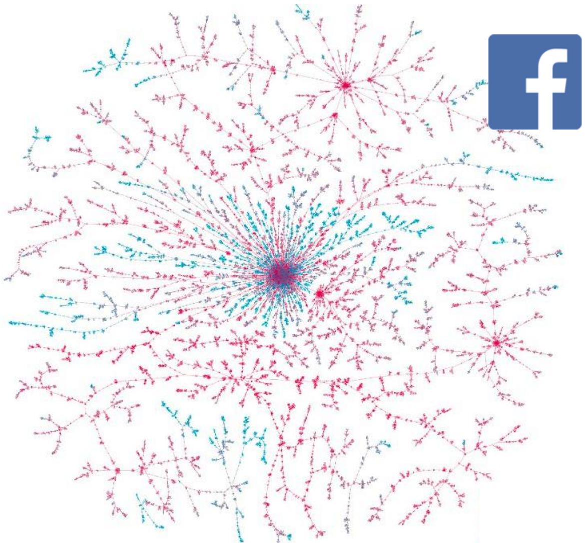


Mobility dynamics

Events are (noisy) observations of a variety of complex dynamic processes…

Example I: Information propagation





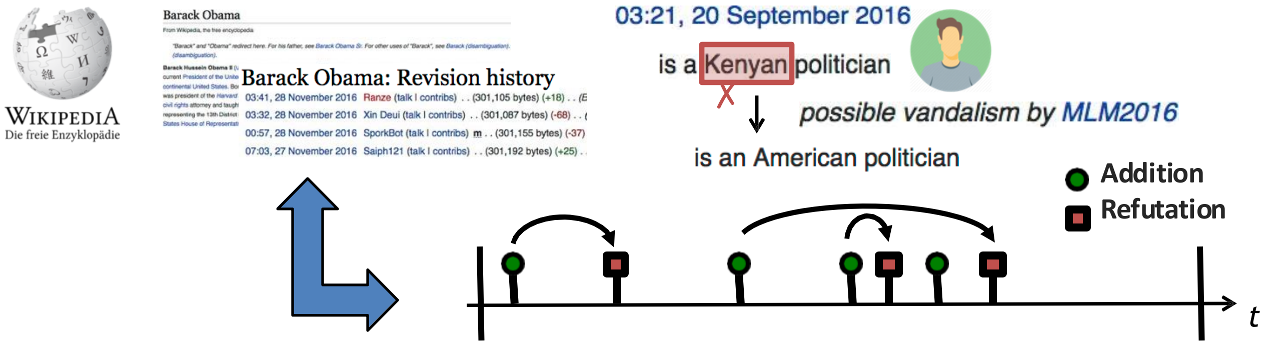
Friggeri et al., 2014

They can have an impact in the off-line world

theguardian

Click and elect: how fake news helped Donald Trump win a real election

Example II: Knowledge creation



Barack Obama

From Wikipedia, the free encyclopedia

"Barack" and "Obama" redirect here. For his father, see Barack Obama Sr. For other uses of "Barack", see Barack (disambiguation). (disambiguation).

Barack Hussein Obama II (
current President of the Unite
continental United States. Bo
was president of the Harvard
civil rights attorney and taugh
representing the 13th District
States House of Representati

Barack Obama: Revision history

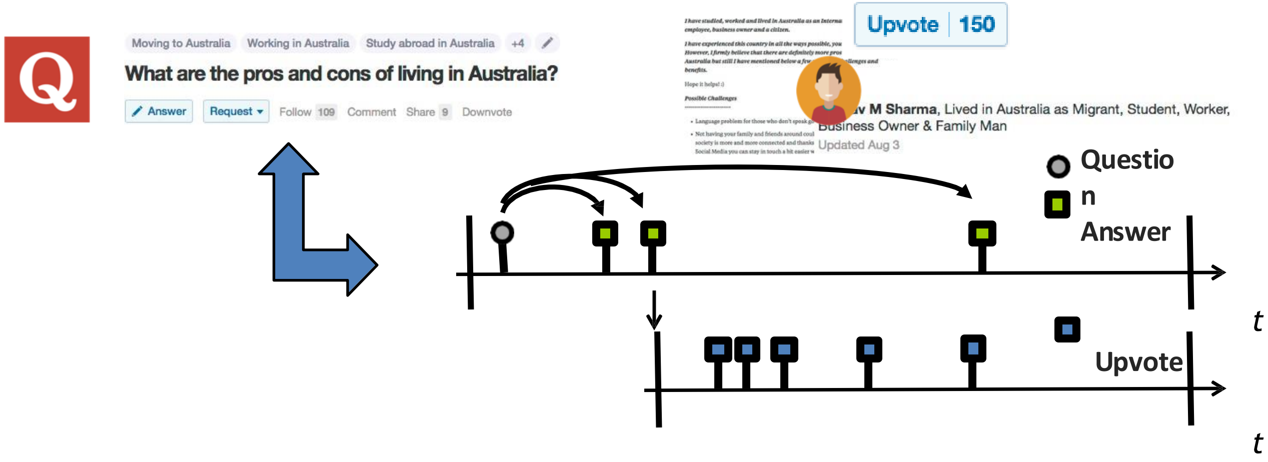
03:41, 28 November 2016 Ranze (talk | contribs) . . (301,105 bytes) (+18) . . (E
03:32, 28 November 2016 Xin Deui (talk | contribs) . . (301,087 bytes) (-68) . . (
00:57, 28 November 2016 SporkBot (talk | contribs) m . . (301,155 bytes) (-37)
07:03, 27 November 2016 Saiph121 (talk | contribs) . . (301,192 bytes) (+25) . .

03:21, 20 September 2016

is a Kenyan politician

possible vandalism by MLM2016

is an American politician



Moving to Australia Working in Australia Study abroad in Australia +4

What are the pros and cons of living in Australia?

Answer Request ▾ Follow 109 Comment Share 9 Downvote

I have studied, worked and lived in Australia as an Interna
employee, business owner and a citizen.

I have experienced this country in all the ways possible, you
However, I firmly believe that there are definitely more pros
Australia but still I have mentioned below a few challenges and
benefits.

Hope it helps! :)

Possible Challenges

- Language problem for those who don't speak go
- Not having your family and friends around coul society is more and more connected and thanks Social Media you can stay in touch a bit easier w

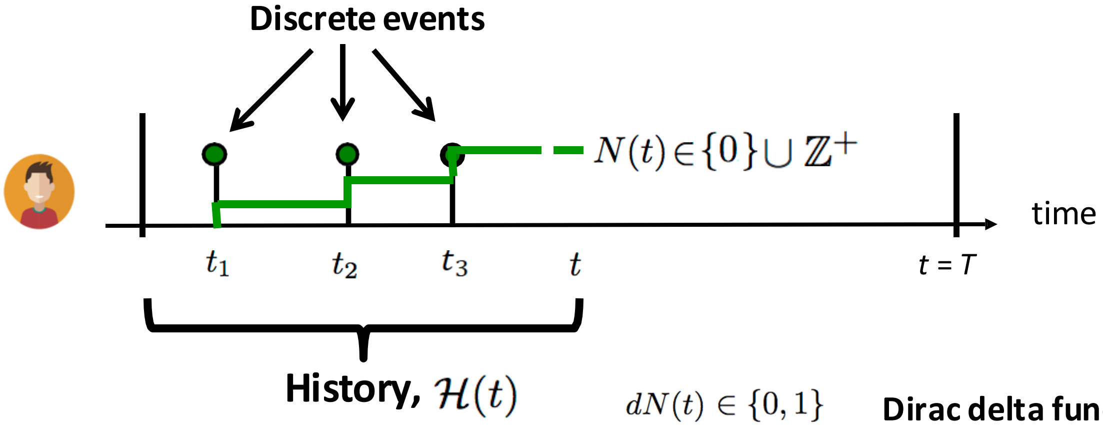
Upvote | 150

v M Sharma, Lived in Australia as Migrant, Student, Worker, Business Owner & Family Man

Updated Aug 3

Temporal point processes

Temporal point process: **A random process whose realization** consists of **discrete events localized in time** $\mathcal{H} = \{t_i\}$

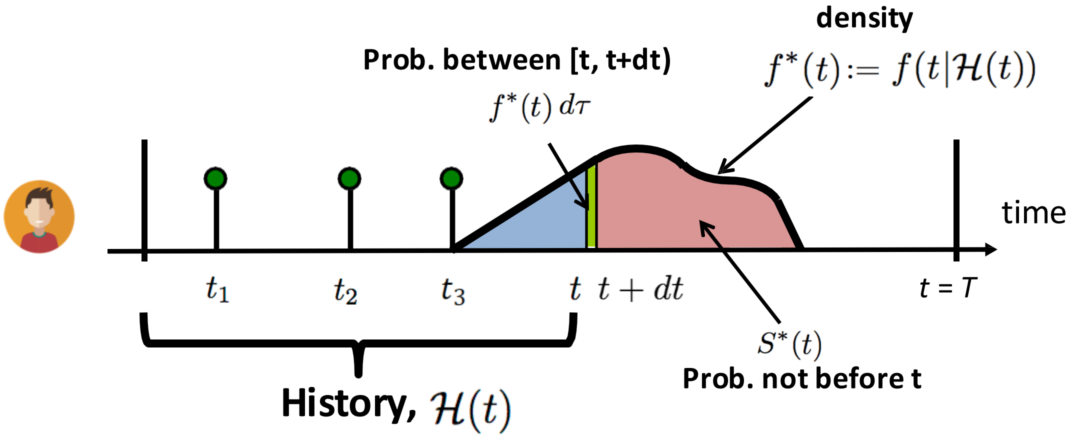


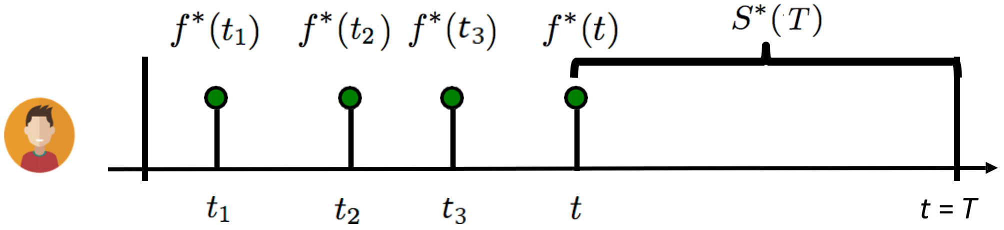
$dN(t) \in \{0, 1\}$

Dirac delta function

Formally: $N(t) = \int_0^t dN(s)$ ⇨ $$dN(t) = \sum_{t_i \in \mathcal{H}} \delta(t - t_i)\, dt$$

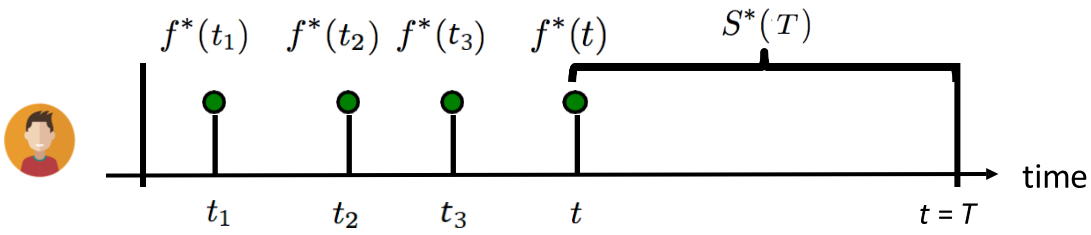
Model time as a random variable

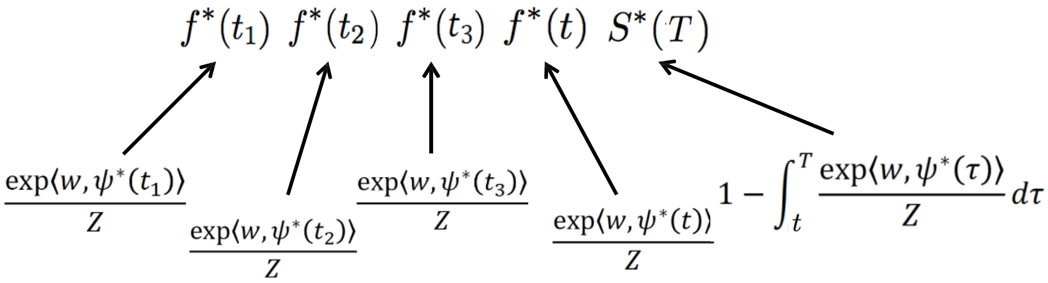




Likelihood of a timeline: $f^*(t_1)\ f^*(t_2)\ f^*(t_3)\ f^*(t)\ S^*(T)$

Problems of density parametrization (I)



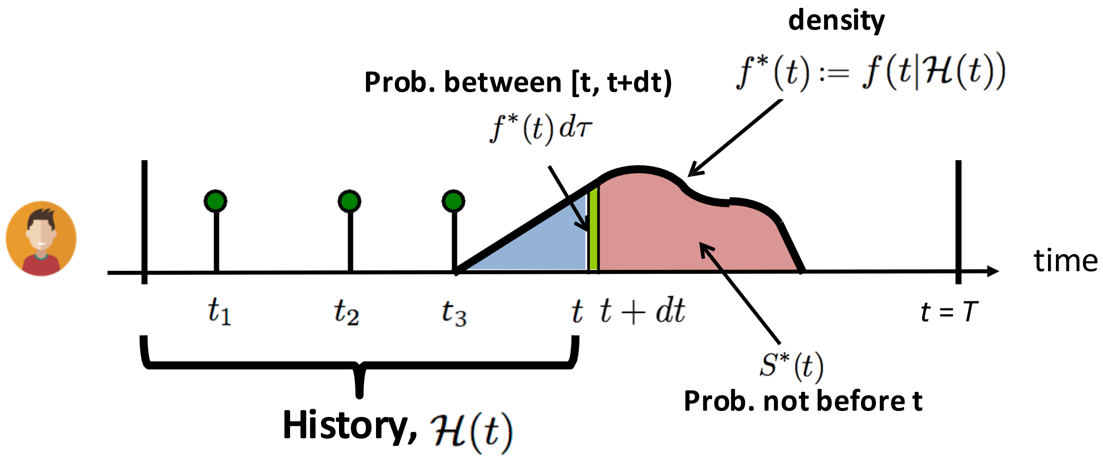


$f^*(t_1)$ $f^*(t_2)$ $f^*(t_3)$ $f^*(t)$ $S^*(T)$
$\frac{\exp\langle w, \psi^*(t_1)\rangle}{Z}$
$\frac{\exp\langle w, \psi^*(t_2)\rangle}{Z}$
$\frac{\exp\langle w, \psi^*(t_3)\rangle}{Z}$
$\frac{\exp\langle w, \psi^*(t)\rangle}{Z}$
$1 - \int_t^T \frac{\exp\langle w, \psi^*(\tau)\rangle}{Z} d\tau$

It is difficult for model design and interpretability:

1. **Densities need to integrate to 1 (i.e., partition function)**
2. **Difficult to combine timelines**

Intensity function



density

$f^*(t) := f(t|\mathcal{H}(t))$

Prob. between [t, t+dt)

$f^*(t)\,d\tau$

time

t_1 t_2 t_3 t $t + dt$ $t = T$

$S^*(t)$

Prob. not before t

History, $\mathcal{H}(t)$

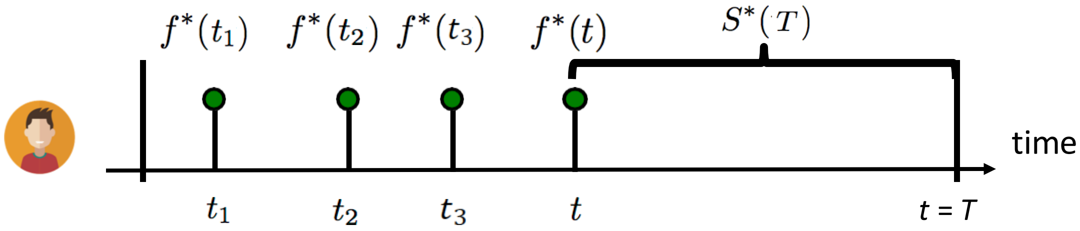
Intensity:

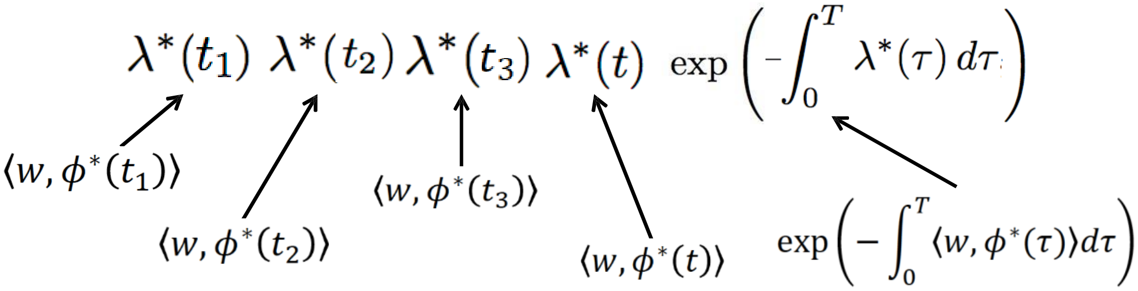
Probability between [t, t+dt) but not before t

$$\lambda^*(t)dt = \frac{f^*(t)dt}{S^*(t)} \geq 0 \quad \Rightarrow \quad \lambda^*(t)dt = \mathbb{E}[dN(t)|\mathcal{H}(t)]$$

Observation: $\lambda^*(t)$ **It is a rate = # of events / unit of time**

Advantages of intensity parametrization (I)





$$\lambda^*(t_1)\ \lambda^*(t_2)\ \lambda^*(t_3)\ \lambda^*(t)\ \exp\left(-\int_0^T \lambda^*(\tau)\, d\tau\right)$$

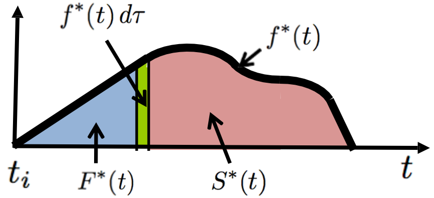
$$\langle w, \phi^*(t_1)\rangle \quad \langle w, \phi^*(t_2)\rangle \quad \langle w, \phi^*(t_3)\rangle \quad \langle w, \phi^*(t)\rangle \quad \exp\left(-\int_0^T \langle w, \phi^*(\tau)\rangle d\tau\right)$$

Suitable for model design and interpretable:

1. **Intensities only need to be nonnegative**
2. **Easy to combine timelines**

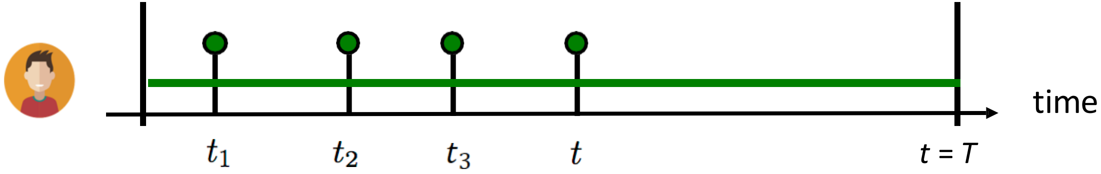
Relation between f*, F*, S*, λ*

$$\begin{tikzcd}
F^*(t) \arrow[rr, shift left, "\frac{dF^*(t)}{dt}"] \arrow[dd, shift left, "1 - F^*(t)"] & & f^*(t) \arrow[ll, shift left, "\int_{t_{i-1}}^{t} f^*(\tau)\, d\tau"] \arrow[dd, bend right, "\frac{f^*(t)}{S^*(t)}"'] \\
& & \\
S^*(t) \arrow[uu, shift left, "1 - S^*(t)"] \arrow[rr, bend left] & & \lambda^*(t) \arrow[uu, "\lambda^*(t)\exp\left(-\int_{t_{i-1}}^{t} \lambda^*(\tau)\, d\tau\right)"'] \arrow[ll, "\exp\left(-\int_{t_{i-1}}^{t} \lambda^*(\tau)\, d\tau\right)"]
\end{tikzcd}$$



Central quantity we will use!

Poisson process



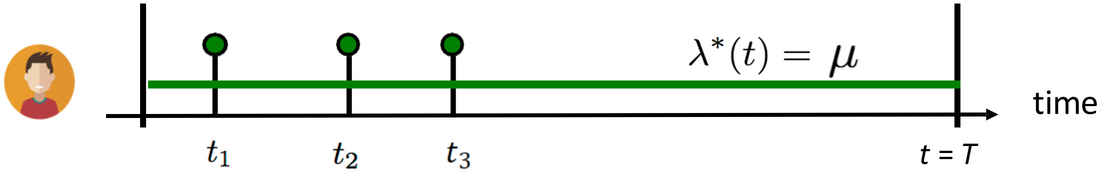
Intensity of a Poisson process

$$\lambda^*(t) = \mu$$

Observations:

1. **Intensity independent of history**
2. **Uniformly random occurrence**
3. **Time interval follows exponential distribution**

Fitting & sampling from a Poisson



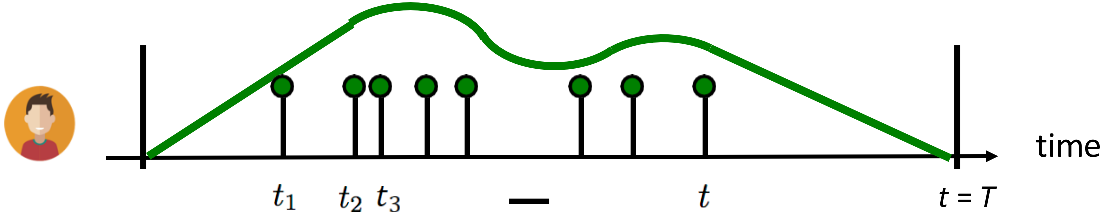
Fitting by maximum likelihood:

$$\mu^* = \underset{\mu}{\text{argmax}}\ 3\log\mu - \mu\, T = \frac{3}{T}$$

Sampling using inversion sampling:

$$t \sim \underbrace{\mu \exp(-\mu(t - t_3))}_{f_t^*(t)} \quad ⇨ \quad t = \underbrace{-\frac{1}{\mu}\log(1 - \overset{\overset{Uniform(0,1)}{\downarrow}}{u}) + t_3}_{F_t^{-1}(u)}$$

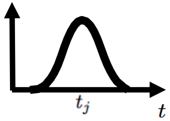
Inhomogeneous Poisson process



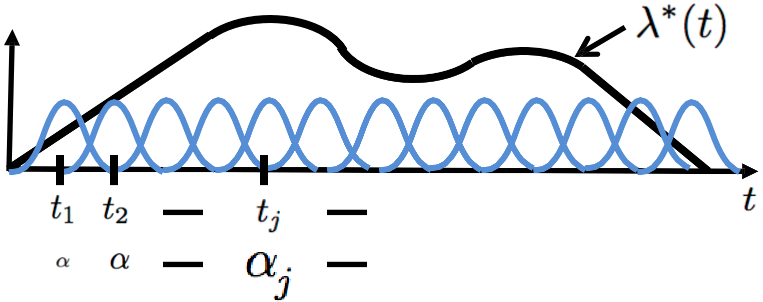
Intensity of an inhomogeneous Poisson process

$$\lambda^*(t) = g(t) \geqslant 0$$ **(Independent of history)**

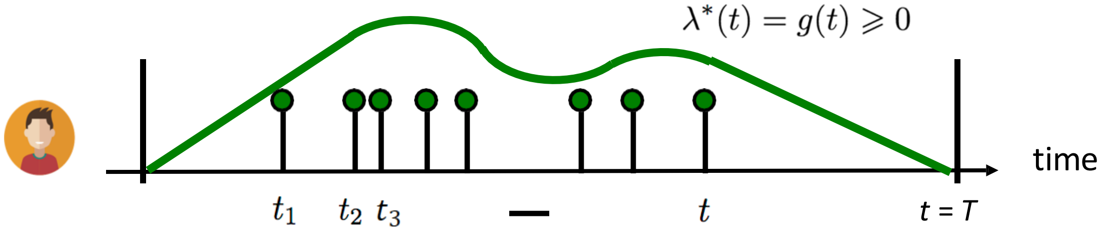
Example:



$$\lambda^*(t) = \sum_j \alpha_j \overbrace{k(t - t_j)}$$



Fitting & sampling from inhomogeneous Poisson



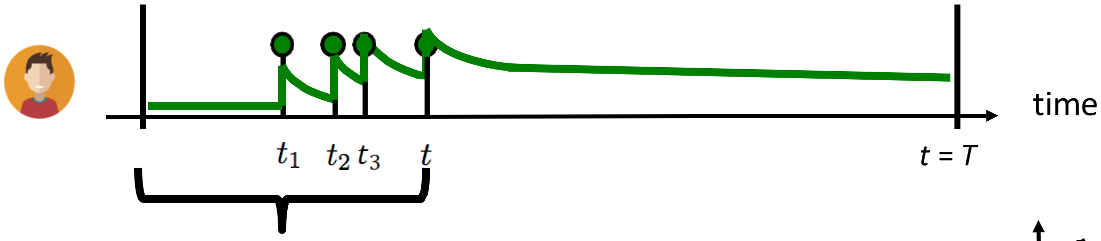
Fitting by maximum likelihood: $\underset{g(t)}{\text{maximize}} \sum_{i=1}^{n} \log g(t_i) - \int_0^T g(\tau)\, d\tau$

Sampling using thinning (reject. sampling) + inverse sampling:

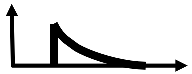
1. **Sample t from Poisson process with intensity μ using inverse sampling**
2. **Generate $u_2 \sim Uniform(0,1)$**
3. **Keep the sample if $u_2 \leq g(t) / \mu$**

$\Big\}$ **Keep sample with prob.** $g(t)/\mu$

Self-exciting (or Hawkes) process



History, $\mathcal{H}(t)$



Intensity of self-exciting (or Hawkes) process:

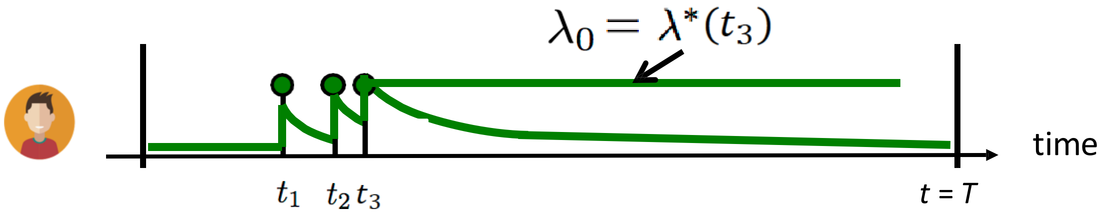
$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}(t)} \overbrace{\kappa_\omega(t - t_i)}^{\text{Triggering kernel}}$$

$$= \mu + \alpha \kappa_\omega(t) \star dN(t)$$

Observations:

1. Clustered (or bursty) occurrence of events
2. Intensity is stochastic and history dependent

Fitting a Hawkes process from a recorded timeline



Fitting by maximum likelihood:

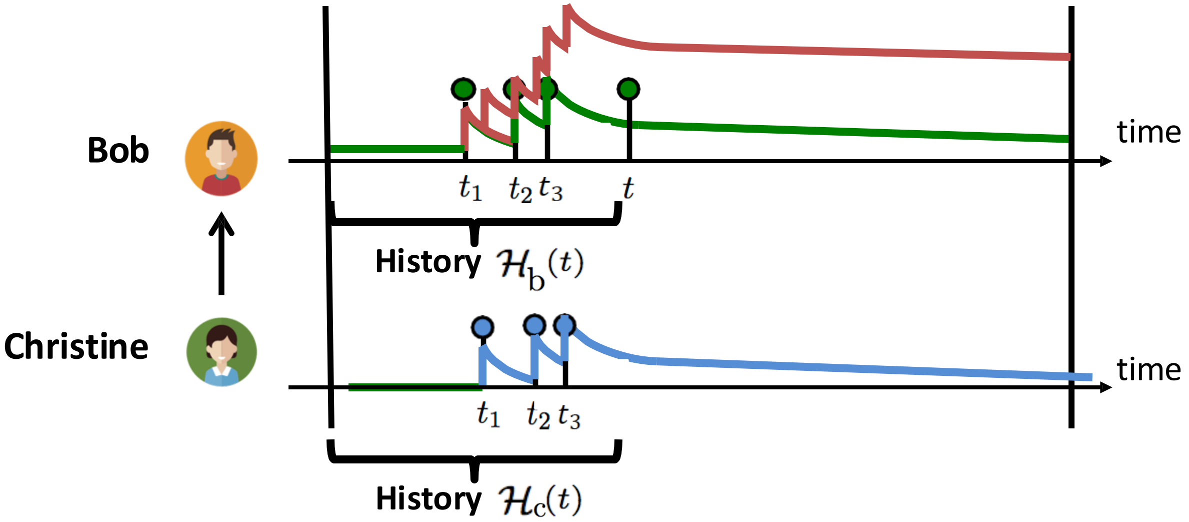
$$\underset{\mu,\alpha}{\text{maximize}} \sum_{i=1}^{n} \log \lambda^*(t_i) - \int_0^T \lambda^*(\tau)\, d\tau$$

The max. likelihood is jointly convex in μ and α

Sampling using thinning (reject. sampling) + inverse sampling:

Key idea: the maximum of the intensity λ_0 changes over time

Mutually exciting process

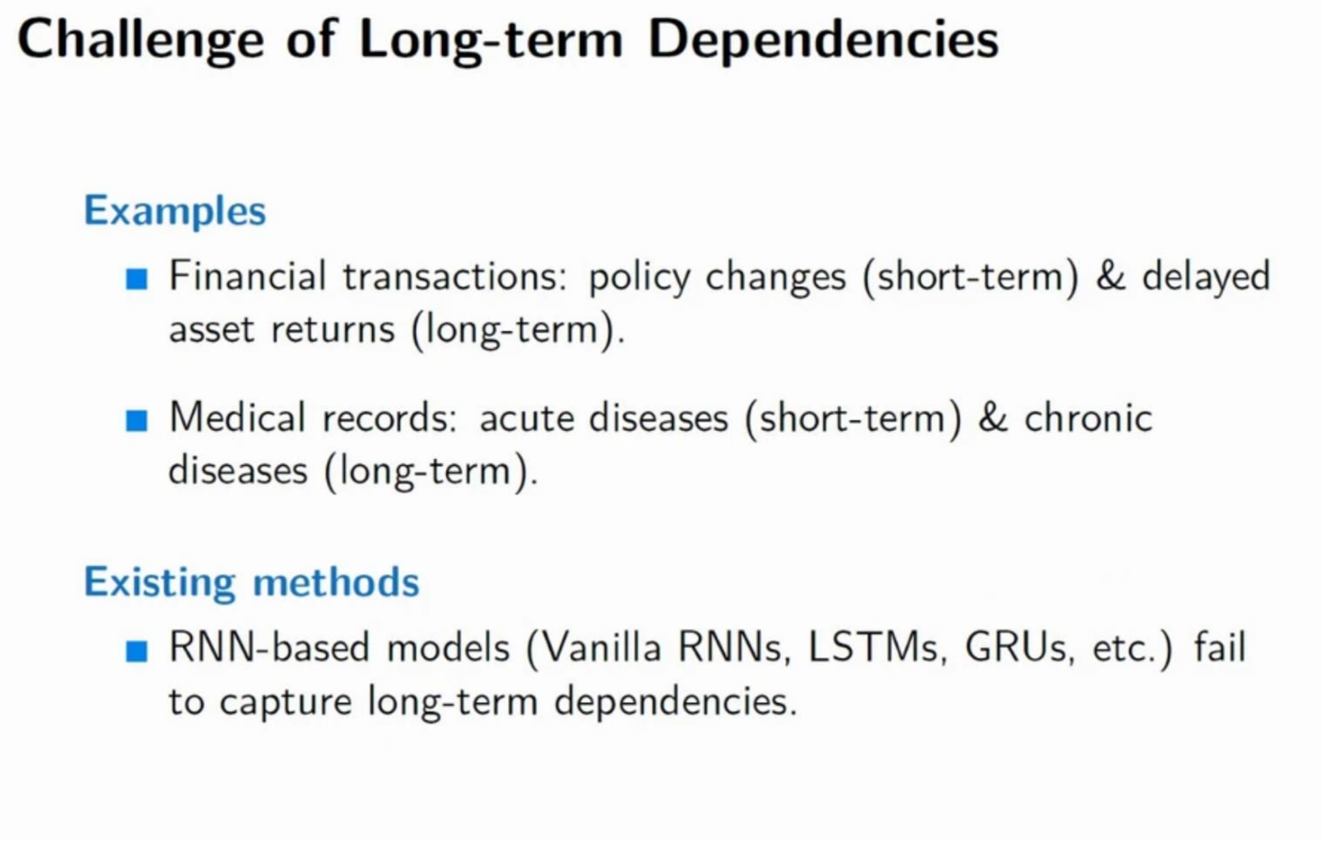


Clustered occurrence affected by neighbors

$$\lambda^*(t) = \mu + \alpha \sum_{t_i \in \mathcal{H}_b(t)} \kappa_\omega(t - t_i) + \beta \sum_{t_i \in \mathcal{H}_c(t)} \kappa_\omega(t - t_i)$$

Deep Temporal Point Process

Deep Temporal Point Process Prediction



Challenge of Long-term Dependencies

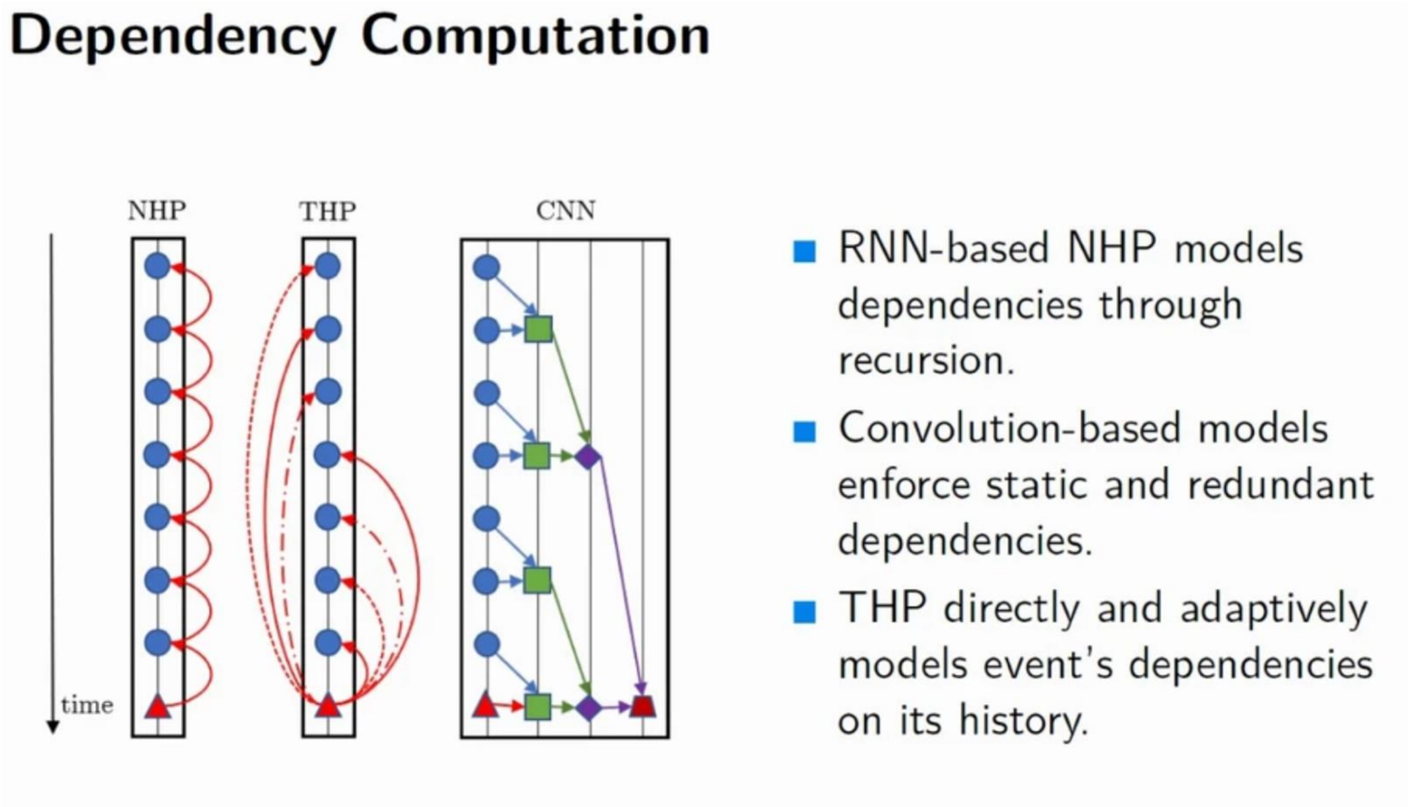
Examples

- Financial transactions: policy changes (short-term) & delayed asset returns (long-term).
- Medical records: acute diseases (short-term) & chronic diseases (long-term).

Existing methods

- RNN-based models (Vanilla RNNs, LSTMs, GRUs, etc.) fail to capture long-term dependencies.

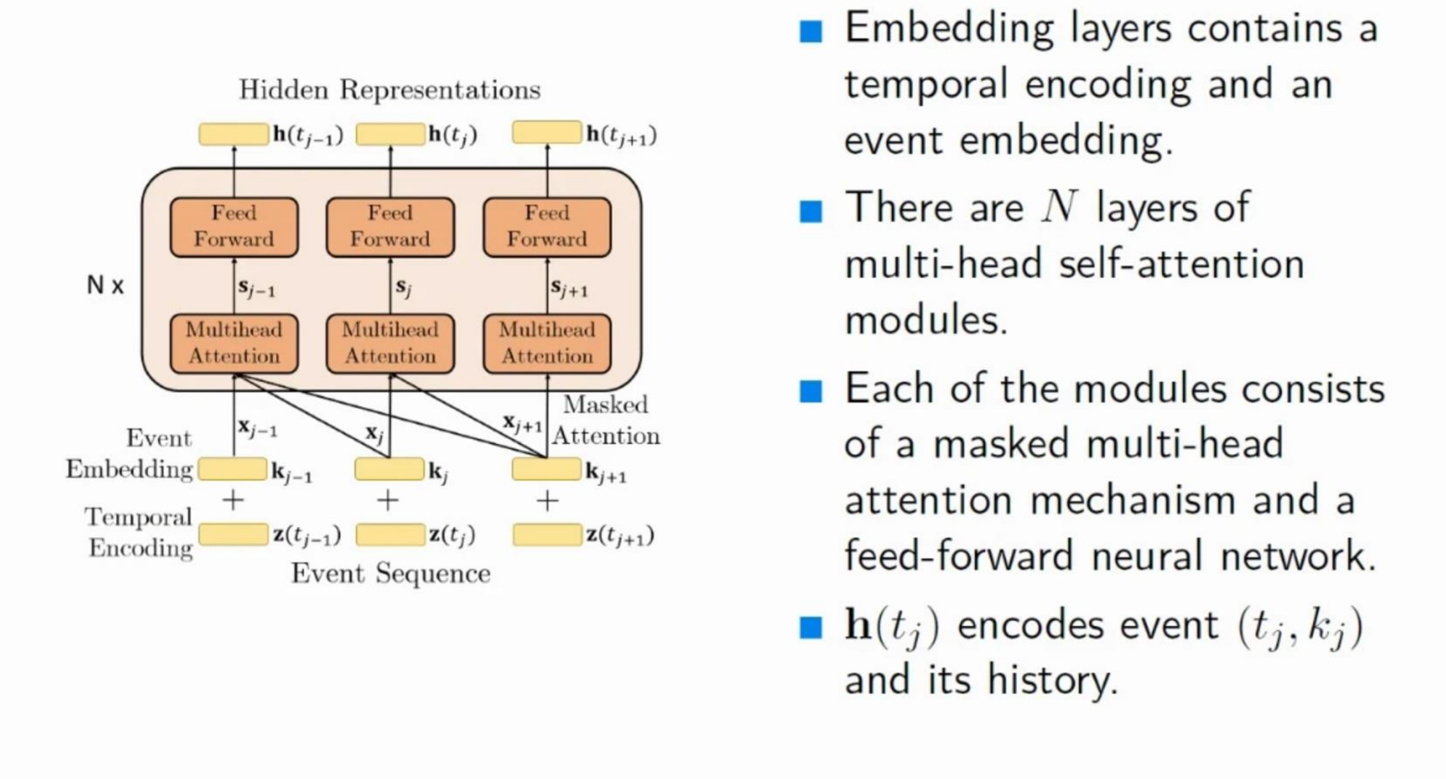
Deep Temporal Point Process Prediction



Dependency Computation

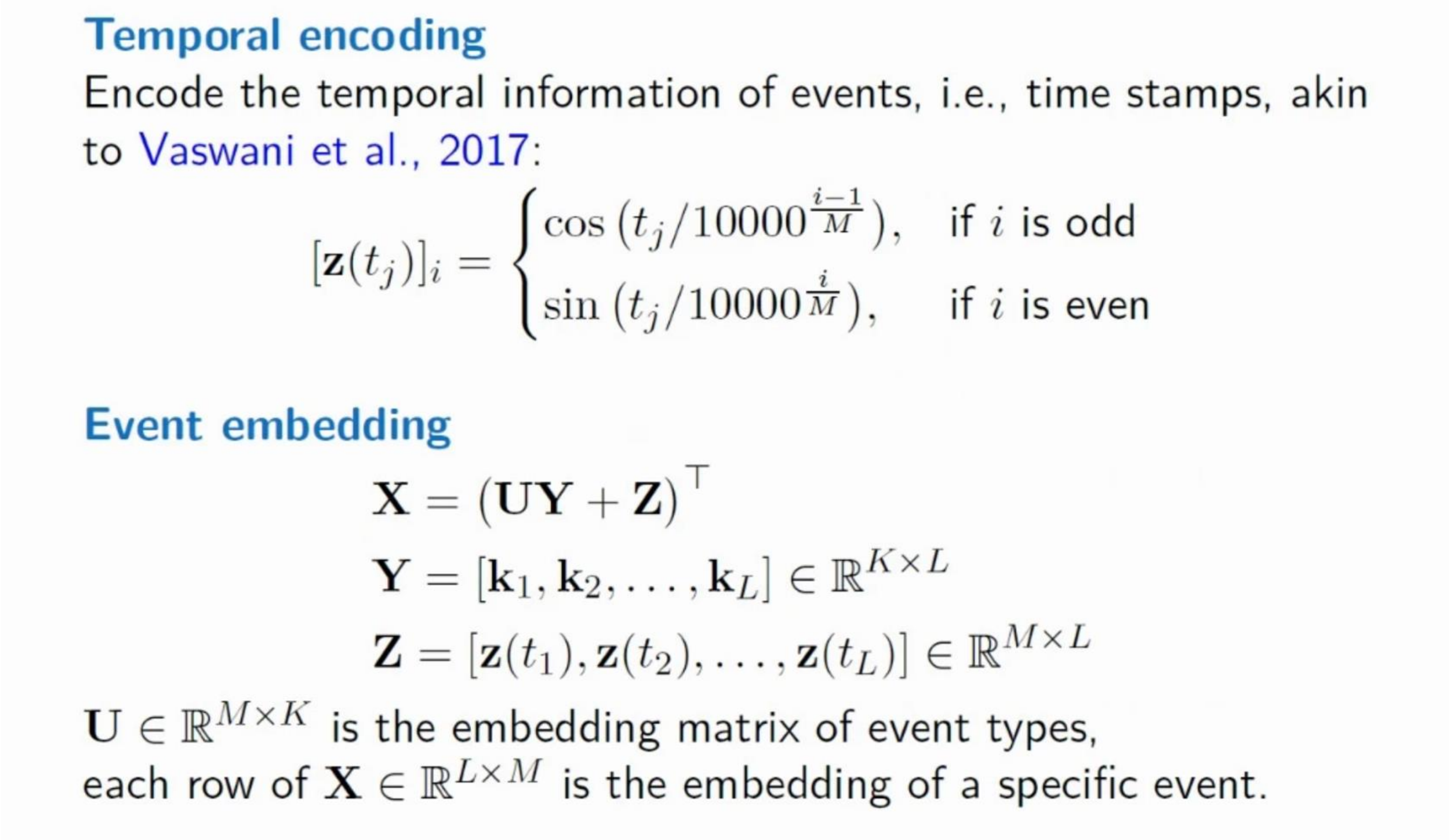
- RNN-based NHP models dependencies through recursion.
- Convolution-based models enforce static and redundant dependencies.
- THP directly and adaptively models event's dependencies on its history.

Transformer Hawkes Process



- Embedding layers contains a temporal encoding and an event embedding.
- There are N layers of multi-head self-attention modules.
- Each of the modules consists of a masked multi-head attention mechanism and a feed-forward neural network.
- $\mathbf{h}(t_j)$ encodes event (t_j, k_j) and its history.

Embedding Layers



Temporal encoding

Encode the temporal information of events, i.e., time stamps, akin to Vaswani et al., 2017:

$$[\mathbf{z}(t_j)]_i = \begin{cases} \cos\left(t_j / 10000^{\frac{i-1}{M}}\right), & \text{if } i \text{ is odd} \\ \sin\left(t_j / 10000^{\frac{i}{M}}\right), & \text{if } i \text{ is even} \end{cases}$$

Event embedding

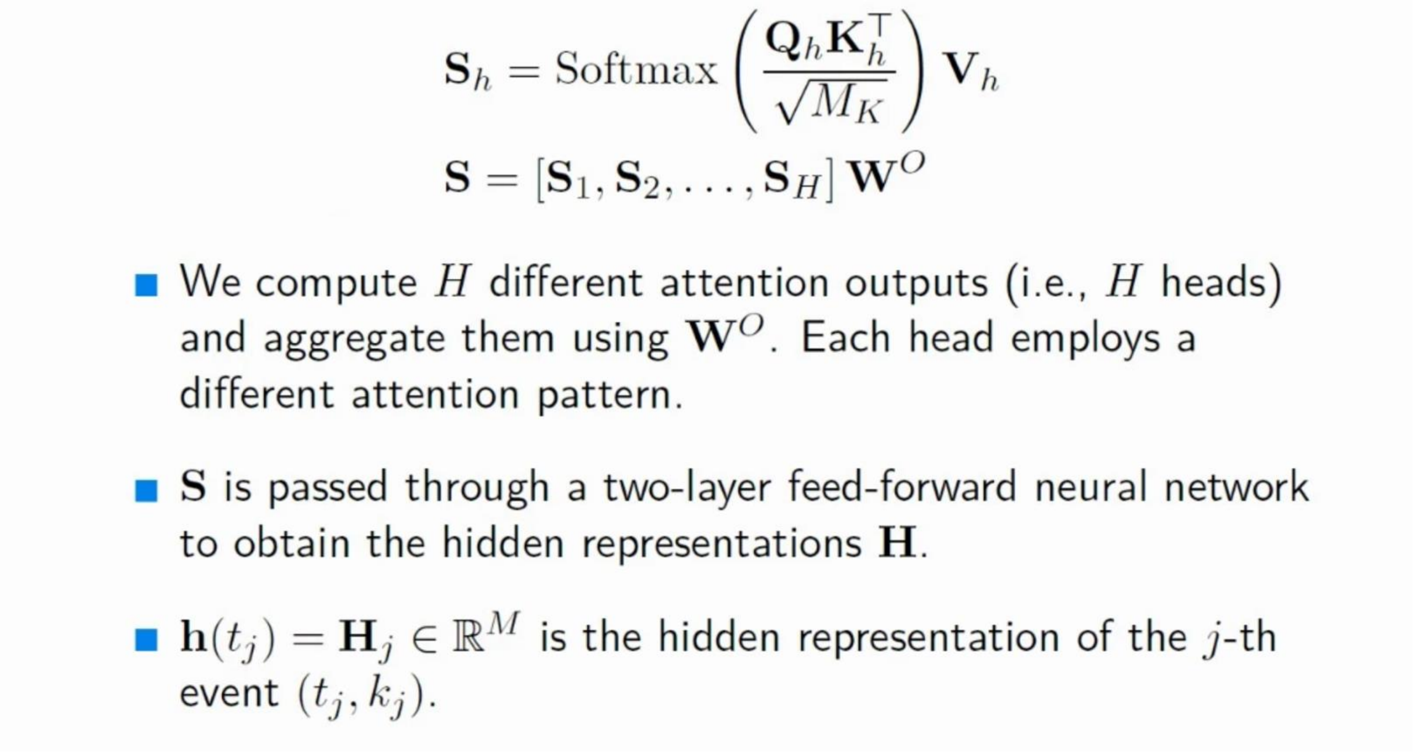
$$\mathbf{X} = (\mathbf{UY} + \mathbf{Z})^\top$$

$$\mathbf{Y} = [\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_L] \in \mathbb{R}^{K \times L}$$

$$\mathbf{Z} = [\mathbf{z}(t_1), \mathbf{z}(t_2), \ldots, \mathbf{z}(t_L)] \in \mathbb{R}^{M \times L}$$

$\mathbf{U} \in \mathbb{R}^{M \times K}$ is the embedding matrix of event types,
each row of $\mathbf{X} \in \mathbb{R}^{L \times M}$ is the embedding of a specific event.

Multi-head Self-attention Modules



$$\mathbf{S}_h = \mathrm{Softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{M_K}}\right)\mathbf{V}_h$$

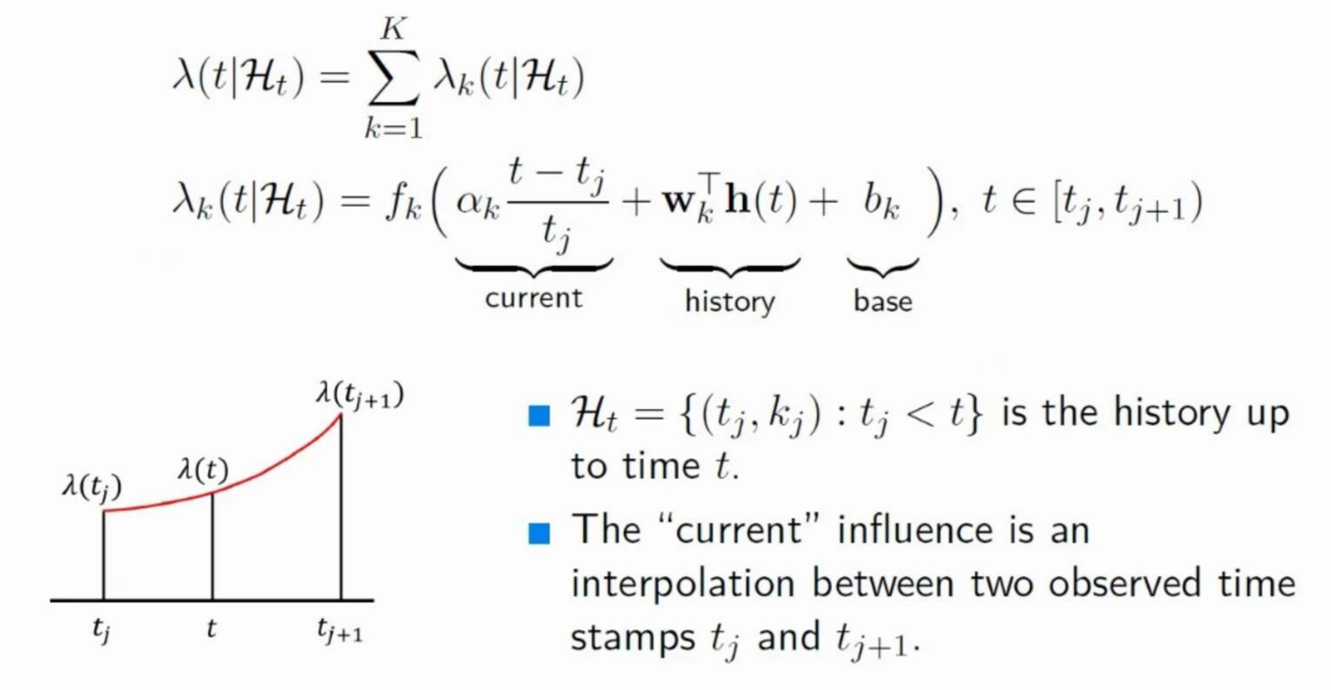
$$\mathbf{S} = [\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_H]\,\mathbf{W}^O$$

- We compute H different attention outputs (i.e., H heads) and aggregate them using $\mathbf{W}^O$. Each head employs a different attention pattern.

- $\mathbf{S}$ is passed through a two-layer feed-forward neural network to obtain the hidden representations $\mathbf{H}$.

- $\mathbf{h}(t_j) = \mathbf{H}_j \in \mathbb{R}^M$ is the hidden representation of the j-th event (t_j, k_j).

Continuous Time Conditional Intensity



$$\lambda(t|\mathcal{H}_t) = \sum_{k=1}^{K} \lambda_k(t|\mathcal{H}_t)$$

$$\lambda_k(t|\mathcal{H}_t) = f_k\Big(\underbrace{\alpha_k \frac{t-t_j}{t_j}}_{\text{current}} + \underbrace{\mathbf{w}_k^\top \mathbf{h}(t) +}_{\text{history}} \underbrace{b_k}_{\text{base}}\Big),\ t \in [t_j, t_{j+1})$$

- $\mathcal{H}_t = \{(t_j, k_j) : t_j < t\}$ is the history up to time t.
- The "current" influence is an interpolation between two observed time stamps t_j and t_{j+1}.

Comparison Results

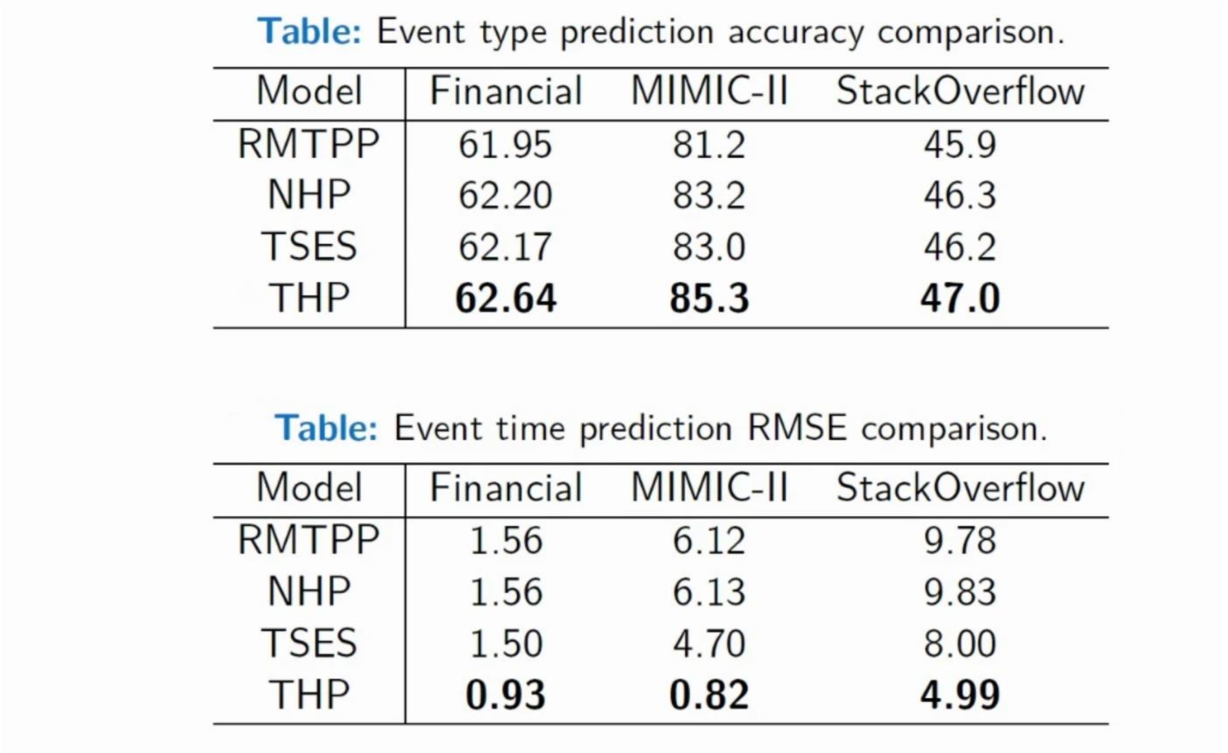


Table: Event type prediction accuracy comparison.

Model	Financial	MIMIC-II	StackOverflow
RMTPP	61.95	81.2	45.9
NHP	62.20	83.2	46.3
TSES	62.17	83.0	46.2
THP	**62.64**	**85.3**	**47.0**

Table: Event time prediction RMSE comparison.

Model	Financial	MIMIC-II	StackOverflow
RMTPP	1.56	6.12	9.78
NHP	1.56	6.13	9.83
TSES	1.50	4.70	8.00
THP	**0.93**	**0.82**	**4.99**

Summary

- Understanding different properties of time series is important
  - Stationarity, Trend, Seasonality, etc.
- Time series forecasting:
  - Identify trend and seasonality, and “remove” it
  - Model the stationary time series: ARIMA
- Deep Time-series forecasting:
  - Transformer-based models dominate – Informer
- Temporal point process models:
  - Event data can be modeled using self-exciting Hawkes process.
  - Transformer Hawkes process – Deep TPP modelling.

Thanks

questions?

Email: sourangshu@cse.iitkgp.ac.in