Time Series Analysis: Models and Methods (A Survey*)

*Some figures and sections are adopted from tutorial by E. Fox in ICML’15

The Data Science and Decisions Lab, UCLA
General Procedure for time series modelling: when, how and why?

1- Forecasting
2- Hypothesis testing
3- Control
4- Clustering
5- Learning structure & dynamics

Pick a model based on the structure of the problem and the validity of assumptions

Validation, statistical efficiency
Medical Problems involving Time Series Analysis

1- Forecasting: -> **Regression analysis**
Predicting future values of physiological measurements, predicting length of a hospital’s waiting list, predicting population level spread of epidemics, survival analysis, etc.

2- Hypothesis testing: -> **Early classification**
Detecting disorders, confounding effects of treatments, testing patients’ latent classes, etc.

3- Control: -> **Planning interventions over time**

4- Clustering: -> **Learning similarities across temporal data**

5- Learning structure & dynamics: -> **Learning causalities**
Medical References involving Time Series Analysis


... and many more...
How to start?

Intuition and understanding the idiosyncrasies of the problem!

- Is the series stationary? Disorders?
- Strong mixing?
- Multiple dependent series?
- Causalities between series?
- Exogenous interventions?

Pick a set of Models

Evaluate and eliminate

The Data Science and Decisions Lab, UCLA
The two dimensions of multivariate time series

Every model captures:

**Evolution**: impacts timeliness of decisions and accuracy of predictions

**Relational structure**: impacts cost, computational and statistical efficiency
1) Autoregressive Moving Average Models (ARMA)

**Assumptions**

Underlying process being modeled is (weakly) stationary: the process exhibits no trends over time (i.e. constant mean and variance).

**Construction: univariate ARMA(p,q) model**

\[
X_t = c + \epsilon_t + \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{i=1}^{q} \beta_i \epsilon_{t-i}
\]

- **Constant**
- **Gaussian noise**
- **AR coefficients**
- **MA coefficients**
1) ARMA: stationarity of the generative model
1) ARMA model capturing evolution
1) Multivariate ARMA model

Evolution described by AR and MA coefficients, what about relational structure?

Construction: multivariate ARMA\((p,q)\) model

\[
X_t = c + \epsilon_t + \sum_{i=1}^{p} A_i \, X_{t-i} + \sum_{i=1}^{q} B_i \, \epsilon_{t-i}
\]

Companion matrix (AR matrix) and MA matrix capture the relational structure

Variant constructions can capture drifts, periodicity, etc. E.g. ARIMA and seasonal ARIMA.
1) Fitting the ARMA model

Learning the parameters of the ARMA\((p,q)\) model (model fitting)

\[ X_t = c + \epsilon_t + \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{i=1}^{q} \beta_i \epsilon_{t-i} \]

Model fitting = find AR, MA coefficients, and number of such coefficients!
1) Fitting the ARMA model

Learning the parameters of the ARMA($p,q$) model (model fitting)

1) Estimate the number of coefficients $p$ and $q$

Use an information criterion to select a model (e.g. Akaike IC and Bayesian IC).

2) Maximum Likelihood Estimation

Estimating the covariance and mean parameters as a function of the AR and MA parameters.
1) Fitting the ARMA model

Learning the parameters of the ARMA(p,q) model (model fitting)

1) Estimate the number of coefficients p and q: use AIC

- Measures relative quality of statistical models for a given set of data.
- Relative estimate of the information lost when a given model is used to represent the process that generates the data.
- Trade-off between the goodness-of-fit of the model and the complexity of that model.
1) Fitting the ARMA model

Learning the parameters of the ARMA(p,q) model (model fitting)

Estimate the number of coefficients p and q: use AIC

\[
AIC = 2pq - 2 \log(\ell(p, q))
\]

MLE for specific p and q

\[
-2\ell(\mu, \phi, \theta, \sigma^2) = n \log 2\pi + \log |\Gamma_n| + (X - \mu)'\Gamma_n^{-1}(X - \mu)
\]

Maximum AIC = Best Model
1) ARMA model: pros and cons

**Advantages**

Stationarity assumption (coefficients are constant over time) leads to statistical efficiency and ease of construction

**Limitations**

Cannot capture complicated or non-stationary dynamics unless upgraded in a way that sacrifices statistical efficiency

**ARMA is usually used in quantitative finance, econometrics, weather forecasting, etc, in order to predict future values of a series**
1) ARMA model: key papers


2) Diffusion models

Assumptions

Underlying process is:
- Continuous time
- Solves a stochastic differential equation (SDE)
- Has a drift component and a diffusion component
- Usually a Markovian process

\[ dX(t) = \mu(X(t), t) \, dt + \sigma(X(t), t) \, dB(t) \]

Drift component
Diffusion component
Wiener process (standard Brownian motion)
2) Diffusion models: depiction

\[ X(t) \]

Stochastic volatility \( \sigma(X(t), t) \)

\[ \mu(X(t), t) \] Drift component (e.g. clinical deterioration)
2) Diffusion models

Advantages

- Models continuous time, some classical results are tractable
- Statistically efficient and non-stationary: only need MLE for drift and stochastic volatility

Disadvantages

- In many cases discrete time models suffice especially if sampler is exogenous. Many SDE problems are tedious and intractable.
- Mostly limited to Markovian processes.
- Hard to model relational structure.

Unexplored by ML community, intensively used in quantitative finance
2) Diffusion models: depiction

Brownian motion is the limit of many ARIMA models!
2) Diffusion models: key papers


3) Hidden Markov Models (HMMs)

Assumptions

States are not observable, and what is observed is a set of outputs that are observed with a different probability given each state.
3) HMM simulation

- Observation is a probabilistic function of the state
- HMM is a **doubly embedded stochastic process**
3) HMM learning

- Learning an HMM is equivalent to learning the state-transition probabilities \((a_{ij})\) and the emission probabilities \((b_{ik})\).
3) HMM learning

- Learning an HMM is equivalent to learning the state-transition probabilities \((a_{ij})\) and the emission probabilities \((b_{ik})\) given an output sequence.

\[
\text{arg max}_{a,b} \ P \left( Y_1, Y_2, Y_3, \ldots, Y_M \mid \{a_{ij}\}, \{b_{ij}\} \right)
\]


3) HMM learning using dynamic programming

Algorithms for learning HMMs

1- **Brute force**: enumerate all output sequences and compute their likelihood -> exponential complexity!

2- **Viterbi algorithm (dynamic programming)**: same idea as Viterbi decoders in convolutional codes.

Used if there is a known state-space that is not observed, and that has a non-injective map to an observed output sequence.
HMMs in a medical context (e.g. Martin’s problem)

Unobserved true clinical status

Observed physiological stream

Learn model by Viterbi, Decoder emits a risk score!
3) HMM control

Markov Chain → MDP → HMM → POMDP

Actions + rewards
3) HMMs: key papers


4) State-space model

Assumptions

- Continuous state-space
- This leads to linearity assumptions for mapping state transitions and states to outputs.

\[
x_t = Ax_{t-1} + e_t \quad e_t \sim N(0, \Sigma) \\
y_t = Cx_t + w_t \quad w_t \sim N(0, R)
\]
Learning task: finding the matrices mapping states to outputs, and transition matrices across states + covariance of Gaussian noise

\[
x_t = Ax_{t-1} + e_t \quad e_t \sim N(0, \Sigma)
\]
\[
y_t = Cx_t + w_t \quad w_t \sim N(0, R)
\]
4) State-space model

High-dimensional time series: Large matrices A and C
Some temporal streams are redundant:

How to reduce the dimensionality and capture dynamics? Redundancy can be time-varying!!
5) Dynamic Latent Factor Models

**Goal:** Embed high dimensional time series into a lower dimensional space and maintain the dynamics

Dynamic functional connectivity (neuroscience) -> Time varying correlations

Models that seek low dimensional embedding of dynamics
5) Dynamic Latent Factor Models

i.i.d Sequence with no dynamics =>

Latent factor model
5) Dynamic Latent Factor Models

\[ \eta_i \sim N_k(0, I) \]

\[ \epsilon_i \sim N_p(0, \Sigma_0) \]

\[ y_i = \Lambda \eta_i + \epsilon_i \]

\[ \Sigma = \Lambda \Lambda' + \Sigma_0 \]

Marginalize latent factors

The Data Science and Decisions Lab, UCLA
5) Dynamic Latent Factor Models

Incorporating dynamics =>

Dynamic latent factor model
5) Dynamic Latent Factor Models

2-dimensional state-space dynamics

5-dimensional observation-space dynamics

\[ \eta_t = \Phi \eta_{t-1} + \nu_t \]

\[ y_t = \Lambda \eta_t + \epsilon_t \]

\[ N_{k}(0, I) \]

\[ N_{p}(0, \Sigma_0) \]
5) Dynamic Latent Factor Models

Correlation pattern is still fixed over time!

\[ y_t \sim N(0, \Sigma) \]

\[ \Sigma = \Lambda \Sigma_\eta \Lambda' + \Sigma_0 \]

Lag covariance = WSS

\[ \Gamma_y(h) = \text{cov}(y_t, y_{t+h}) = \Lambda \Gamma_\eta(h) \Lambda' \quad h > 0 \]
5) Dynamic Latent Factor Models

Complex dynamics => Nonparametric latent factor evolution
5) Dynamic Latent Factor Models

Complex dynamics =>

Nonparametric latent factor evolution

Usually a Gaussian process is used for nonparametric evolution

$$\eta_t = \psi(x_t) + \nu_t$$

$$y_t = \Lambda \eta_t + \epsilon_t$$

$$N_k(0, I)$$

$$N_p(0, \Sigma_0)$$
What is a Gaussian Process?

- Distribution over functions \( f \sim \text{GP} \ (m, K) \)
- \( m \): mean function, \( K \): covariance Kernel

**Brownian motion is a GP with Matern Kernel**

\[ \kappa(x, x') = \sigma_f^2 \exp \left( -\frac{1}{2\ell^2} (x - x')^2 \right) \]
5) Dynamic Latent Factor Models

Fit the latent factor process using GP regression

\[ \eta_t = f(\eta_{1:t-1}) + \nu_t \]

\[ N_k(0, I) \]

Evolution of latent factors

\[ y_t = \Lambda \eta_t + \epsilon_t \]

\[ N_p(0, \Sigma_0) \]
We modeled **EVOLUTION** by a GP, but How to capture the changing correlation pattern (**RELATIONAL STRUCTURE**)?
5) Dynamic Latent Factor Models

Model every entry in Latent factor matrix by a GP

\[ \Sigma(x) = \Lambda(x)\Lambda(x)' + \Sigma_0 \]
5) Dynamic Latent Factor Models

GP to model evolution and dynamics

\[ \eta_t = f(\eta_{1:t-1}) + \nu_t \]

Evolution of latent factors

GP to model latent factor process

\[ y_t = \Lambda \eta_t + \epsilon_t \]

\[ N_p(0, \Sigma_0) \]
Conclusion: Dynamic Latent Factor Model can capture complicated dynamics and encapsulates most of other models
5) Dynamic Latent Factor Model: key papers


Model Selection

Dataset

Mean function 1
Covariance Kernel 1

Mean function 2
Covariance Kernel 2

... ... ...

Mean function M
Covariance Kernel M

Model fitting
And parameter estimation

AIC

Select best model with a desired model confidence interval
Further topics to discuss....

- **Dynamic Bayesian networks**: modeling cause and effect
- **Real-time Cox regression**: survival analysis based on time series
- **Structure learning**: learning graphs of patients, treatments, etc, based on time series
- **Time series clustering**: personalization!