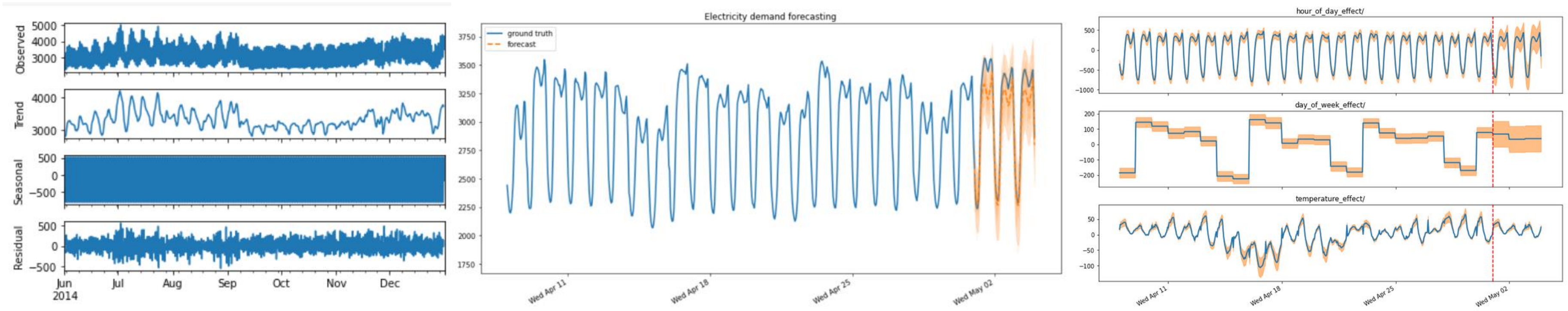
Data Driven Engineering II: Advaced Topics

State Space Models II

Institute of Thermal Turbomachinery
Prof. Dr.-Ing. Hans-Jörg Bauer



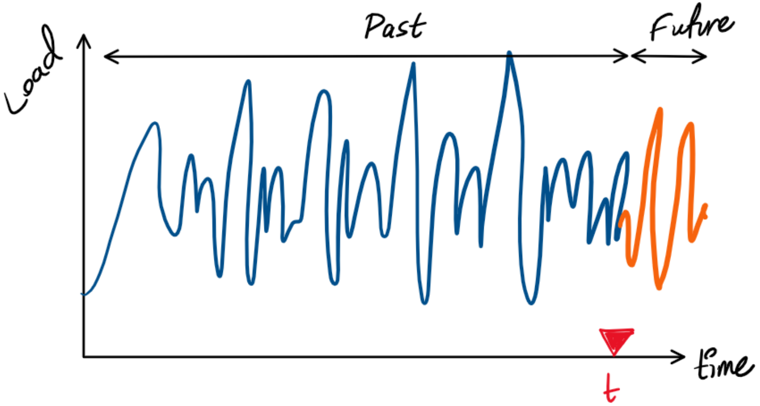
Dynamical Datasets: Time Series

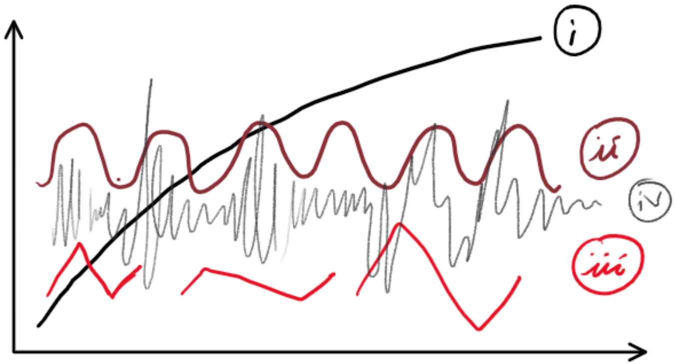
Outline

* Time Series: Overview
* Statistical Models for time series
* State space models ⇒ DDE II
* Machine Learning Part I
* Machine Learning Part II

Time Series Analysis

* Components of time series





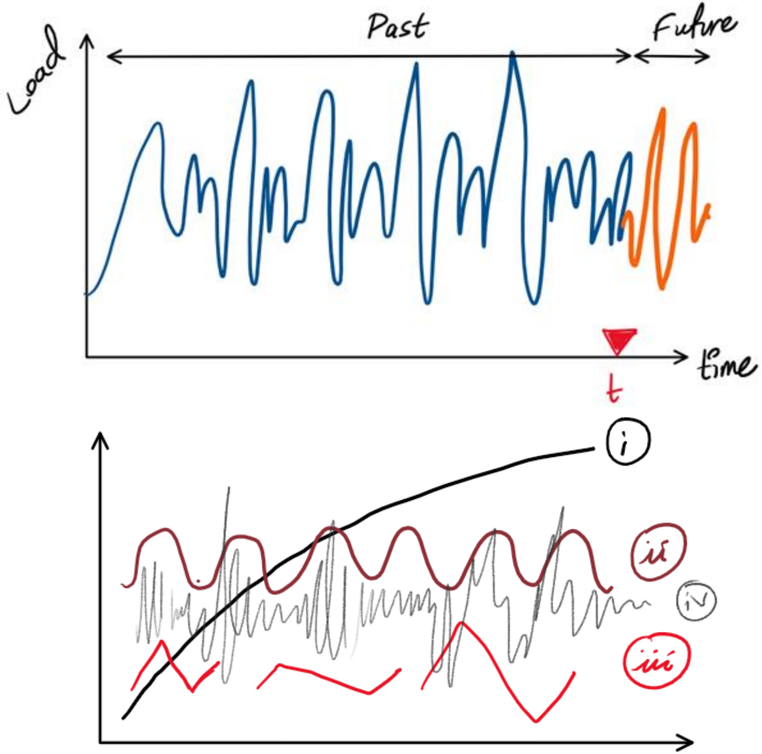
i) Long term trends

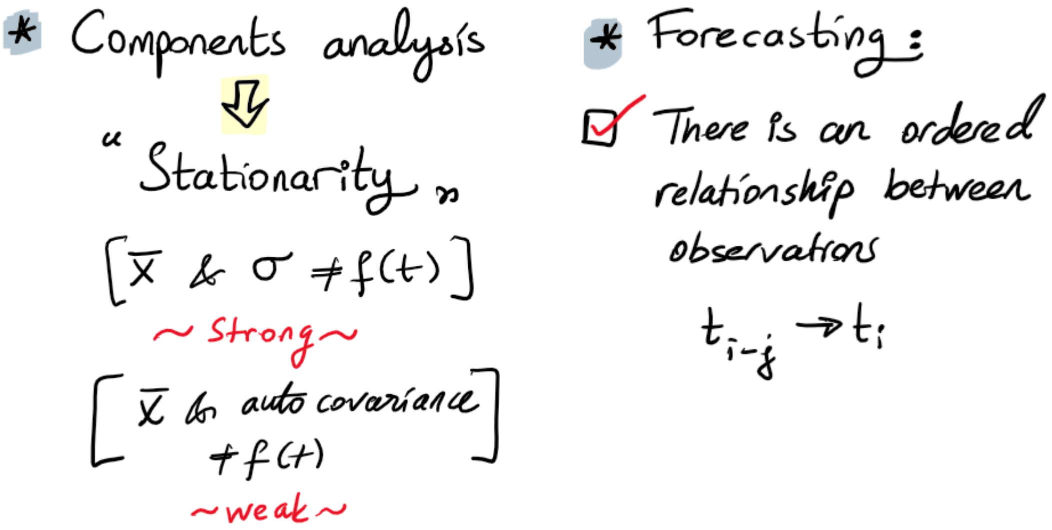
ii) ST seasonal variations

iii) Cyclic variations

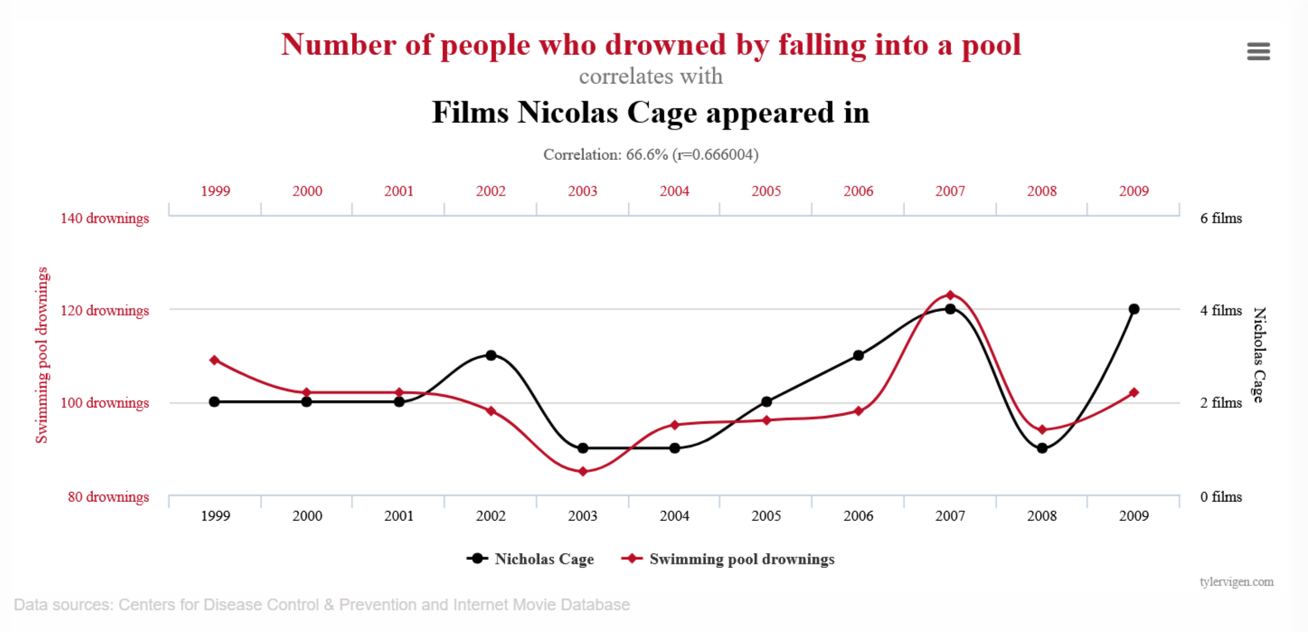
iv) Random fluctuations

Time Series Analysis

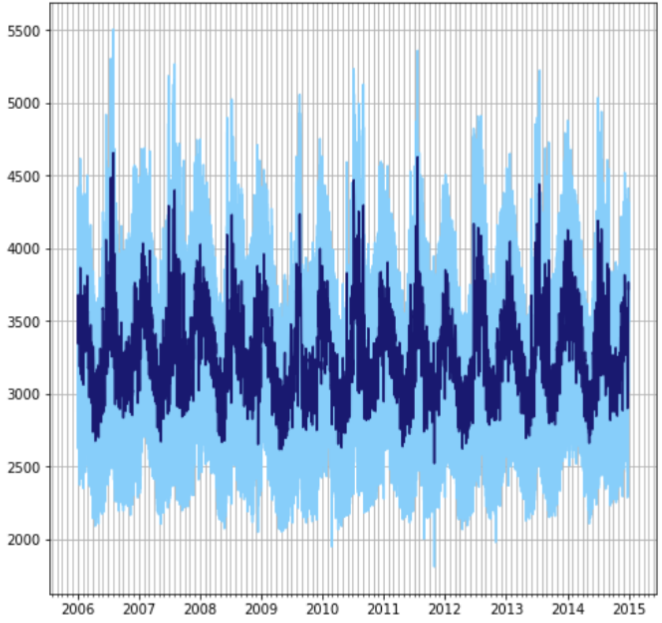




Spurious Correlations



Case: Energy Demand Forecasting



* 8 years data of Temp & Load ($\Delta t = hr$)

? Power Demand forecasting

Short Term

Long Term

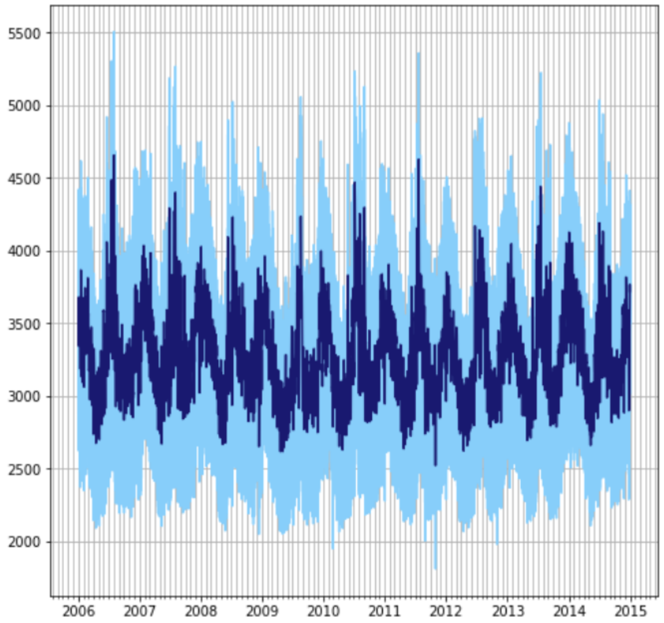
Case: Energy Demand Forecasting

* Short term load forecasting : ~1 hr to 24 hr ⇐ near past is used
  ~demand / supply ⇐ Temperature is an important feature

* Long term LF : ~1 week to months } Planning & investment ⇐ Seasonal patterns
  ~ years ⇐ Long term trends
  ⇐ Climate Models

Case: Energy Demand Forecasting

Typical	STLF	LTLF
Horizon	1 hr – 2 days	≥ 1 months
Granularity	~ hr	~ hr – day
History Range	~ 2 years	~≥ 5 years
Accuracy	≤ 5% error	≤ 25% error
Forecasting freq.	~hr to day	≥ month



Data Exploration: Temporal Nature of data

① How to handle "time stamps"

	Date	Hour	load	T
0	01/01/2004	1	NaN	37.33
1	01/01/2004	2	NaN	37.67
2	01/01/2004	3	NaN	37.00
3	01/01/2004	4	NaN	36.33
4	01/01/2004	5	NaN	36.00

⇨

	load	T
2012-01-05 00:00:00	3167.0	19.00
2012-01-05 01:00:00	3014.0	22.33
2012-01-05 02:00:00	2921.0	22.33
2012-01-05 03:00:00	2874.0	22.00
2012-01-05 04:00:00	2876.0	21.67

Data Exploration: Temporal Nature of data

② Temporal data decomposition → Trend, Seasonality, Noise

Stationarity

⇓

how 'stable' your system

- ☑ Intuition
- ☐ Tests

⇓

how much we should expect the past reflects itself on future?

⇒ "Self Correlations"

Bayesian Structural Time Series

* BSTS $\Leftrightarrow$ linear Gaussian Model << Kalman F.

* Idea: $\left. \begin{matrix} f_1 & f_2 & \\ & & f_3 \\ f_4 & f_i & \end{matrix} \right\}$ 'f, := Hypothesis library

$X = [\, . \; . \; . \,]$ $\rightleftarrows$ $\boxed{f}$ → Decribes & predicts system behavior

... Fitting ...

Bayesian Structural Time Series

* **General psudo-algorithm:**

  (1) Define a structural model

  (2) Specify priors.

  (3) Update estimates of states based on observation data

  (4) Perform variable selection (eg. spike & slab method)

  (5) Apply Bayesian model averaging

  (6) Forecasting

How fitting works?

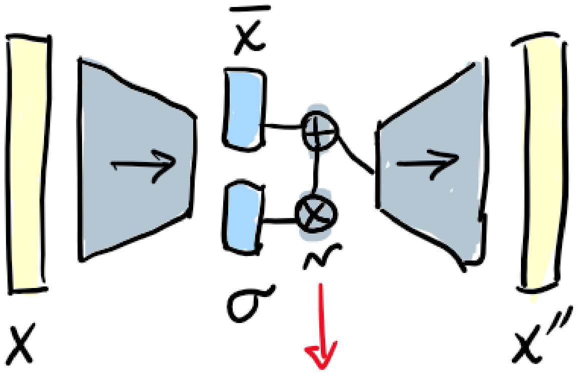
* Minimize Kullback – Leibler divergence (see VAE, DDE-I)

Variational Encoders:

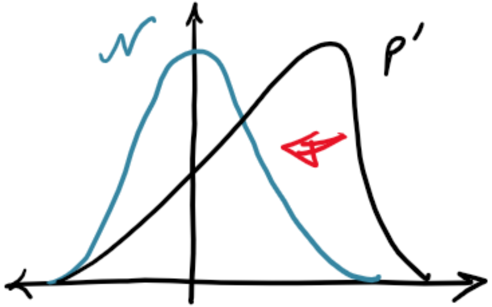
* Gaussian Sampling ⇒ [enforced via cost function]

$\underbrace{\text{[enforced via cost function]}}$

Kullback – Leibler (KL) Divergence ($\mathcal{N}$ & p')



Sampled Randomly ($\mathcal{N}$)



$$D_{KL}\left[p'(\bar{x},\sigma) \,||\, \mathcal{N}(0,1)\right]$$

- $\mathcal{L}_{KL} = -\frac{1}{2}\sum_i 1 + \gamma_i - \exp(\gamma_i) - \bar{x}_i^2$
- $\gamma_i = \ln(\sigma_i^2)$

How fitting works?

* Minimize Kullback–Leibler divergence (see VAE, DDE-I)

- $x :=$ observable ; $z :=$ 'latent' variables ;

- $$p(z|x) = \frac{p(x|z)}{p(x)} \Rightarrow p(x) = \int p(x,z)\,dz \;?$$

- Variational Inference $:=$ $$q^*(z) = \min_{\underbrace{q \in \mathcal{D}}_{\text{candidate approximations}}} KL\big(q(z)\,\|\,p(z|x)\big)$$

How fitting works?

- $$KL\left(q(z) \| p(z|x)\right) = \underbrace{E\left[\log(q(z))\right] - E\left[\log(p(z|x))\right]}_{(-ELBO):\ \text{evidence lower band}} + \log(p(x))$$!

$$ELBO(q) = E\left[\log(p(z,x))\right] - E\left[\log(q(z))\right]$$

$$= E\left[\log(p(x|z))\right] - KL\left(q(z) \| p(z)\right)$$

~ balance btw. expected likelihood ~ prior

How fitting works?

- Variational Inference $:= q^*(z) = \underbrace{\min_{q \in \mathcal{D}}}_{\text{candidate approximations}} \text{ELBO}(q)$

Tensorflow

* Create variational posteriors;

- Sample prior distributions
- Select an optimizer
- Fit posteriors by min. ELBO

```python
# Building the variational surrogate posteriors `qs`:
#https://www.tensorflow.org/probability/api_docs/python/tfp/sts/build_f
variational_posteriors = tfp.sts.build_factored_surrogate_posterior(
    model=load_model)
```

```python
num_variational_steps = 120
optimizer = tf.optimizers.Adam(learning_rate=.1)
```

```python
elbo_loss_curve = tfp.vi.fit_surrogate_posterior(
  target_log_prob_fn=load_model.joint_log_prob(
      observed_time_series=train),
  surrogate_posterior=variational_posteriors,
  optimizer=optimizer,
  num_steps=num_variational_steps)
```

colab

Markov Chain Monte Carlo (MCMC)

* invented after MC @ Los Alamos (1953)

  ↳ liq. phase eq. with its vapor (Eq. Thermodynamics)

  ↳ Metropolis Algorithm

  ↳ M-Hasting

  ↳ Gibbs Sampler

} Simulates posterior dist.

Enables full Bayesian interface

Markov Chain Monte Carlo (MCMC)

* Markov Chains ⇒ early 20^{th} Cent; alternations of letters in a poem; "Onegin" by Poeshkin

- Characterization of Sequence of random variables;

- $X_1, X_2, X_3, X_4 \dots X_n, X_{n+1}$ $\left\{ \begin{array}{l} (n+1) \text{ depends only} \\ (n) \end{array} \right.$

Markov Chain Monte Carlo (MCMC)

* Hidden Markov Models $\Leftrightarrow$ Kalman Filter
  - variables we observe may not be the most descriptive
  - "unsupervised learning"
    - ↳ we have an idea about how system works.

Eg

State	L	G
L	0.6	0.4
G	0.2	0.8

? (0) state → L
(1) 60% ⇒ L
40% ⇒ G

Markov Chain Monte Carlo (MCMC)

(Hidden) X_1 X_2 X_3 X_4 X_5 X_n

$\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$

(Observ.) y_1 y_2 y_3 y_4 y_5 y_n

$\lambda(\pi_{ij}, b_{ij}, a_{ij})$

$\Updownarrow$

y_i

* We can describe a system if we know:

- initial state of the system (π_{ij})
- emission probability; prob. of observing y_i given X_i (b_{ij})
- Transition probability; $X_i \rightarrow X_{i+n}$ (a_{ij})

iterative solution for λ

Markov Chain Monte Carlo (MCMC)

Karlsruhe Institute of Technology

(Hidden) X_1 X_2 X_3 X_4 X_5 X_n

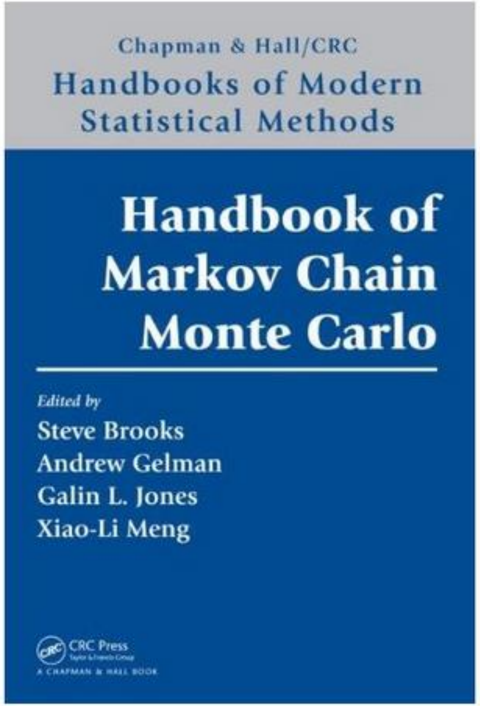
(Observ.) y_1 y_2 y_3 y_4 y_5 y_n

$\lambda\,(\pi_{ij},\ b_{ij},\ a_{ij})$

λ — Frequentist inference — "with enough sampling, we can find it"
- Expectation Maximization (EM)
- ...
- Baum–Welch Algorithm

λ — Bayesian inference — "continuously update the posterior for best λ"
- MCMC
- Gibbs Sampling
- VT Algorithm


Ateliers & Saveurs in Montreal



Variational Inference: A Review for Statisticians

David M. Blei
Department of Computer Science and Statistics
Columbia University

Alp Kucukelbir
Department of Computer Science
Columbia University

Jon D. McAuliffe
Department of Statistics
University of California, Berkeley

May 11, 2018

Additional Notes