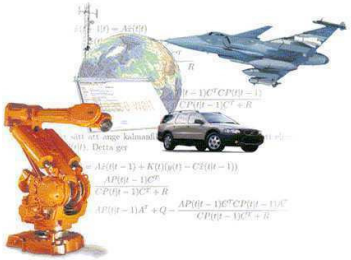
Data Science: From System Identification to (Deep) Learning and Big Data



Lennart Ljung

Reglerteknik, ISY, Linköpings Universitet

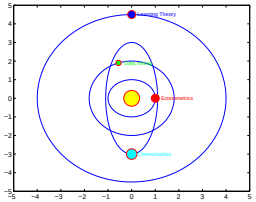
Bottom Line

- Many Buzzwords: Data Science, Machine Learning, Deep Learning, Big Data ...
- One well-defined and well-established Research Area: System Identification
- How does the new fancy development relate to such a classical area?

Data Science?

Wikipedia: *Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, similar to data mining.*

- "Extract knowledge from data": All of science!
- Many research fields have been developed around this in different scientific disciplines, and an extensive and diverse nomenclature exists over the areas.
- But a handful of concepts exist that are the core of the topic.



A picture: There is a core of central material, encircled by the different communities. (statistics, system identification, econometrics, chemometrics, machine learning ...)

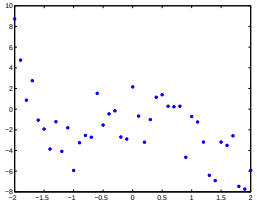
The core

Central terms

- Model $\mathbf{m}$ ("extracted knowledge")
- Model Set/Model Structure $\mathcal{M}$ – Complexity (Flexibility) $\mathcal{C}$
- Information – Data Z
- Estimation – Validation (Learning – Generalization)
- Model fit $\mathcal{F}(\mathbf{m}, Z)$: Agreement between model output and measured data

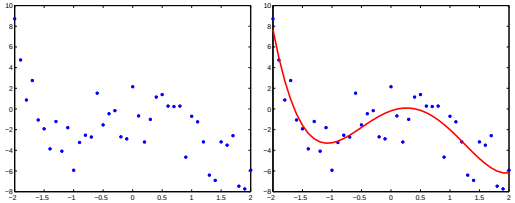
Estimation

information in data



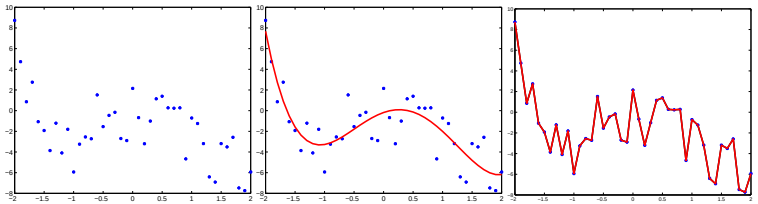
Estimation

Squeeze out the relevant information in data



Estimation

Squeeze out the relevant information in data. (BUT NOT MORE!)



All data contain Information and Misinformation (“Signal and noise”).

So need to meet the data with a prejudice!

Estimation Prejudices

Nature is Simple!

Occam’s razor, Lex Parsimoniae... (Willam of Ockham, $\sim$ 1300 AD)

God is subtle, but He is not malicious (Einstein)

So, conceptually:

$$\hat{\mathfrak{m}} = \arg\min_{\mathfrak{m}\in\mathcal{M}} \text{(Fit + Complexity Penalty)}$$

Estimation and Validation

"Learning and Generalization"

- When the fit beween model output and measured output has been optimized, we see how well the model could reproduce the estimation data.
- If many parameters have been used, this fit can be quite good. Do not be impressed by by a good fit to data in a flexible model set!
- The real test is to see how well the model can reproduce a new, independent data set - the Validation Data

Bias and Variance

$\mathcal{S}$ – True system $\mathcal{M}$– Model Structure $\hat{\mathfrak{m}}$ – Estimate

$$\mathfrak{m}^* = E\hat{\mathfrak{m}} \approx \arg\min_{m\in\mathcal{M}} \text{fit}(\mathcal{S} - m)$$

Then

$$E\|\mathcal{S} - \hat{\mathfrak{m}}\|^2 = \|\mathcal{S} - \mathfrak{m}^*\|^2 + E\|\hat{\mathfrak{m}} - \mathfrak{m}^*\|^2$$

MSE = B: BIAS + V: Variance

Error: = Systematic + Random

$\hat{\mathfrak{m}} \in \mathcal{M}$: As $\mathcal{C}(\mathcal{M})$ increases, B decreases &V increases

This bias/variance trade-off is at the heart of estimation.

Note that the $\mathcal{C}$ that minimizes the MSE typically has a B$\neq 0$!

Core Summary

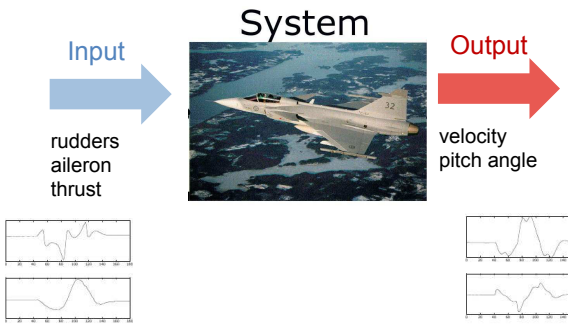
- Data: Could be signals, texts, images,....
- Model set: Mapping from observed data to a result variable ("output")
- Estimation: Optimize fit between estimation data and model output
- Validation: Gain confidence in estimated model: Typically by testing in on new ("validation") data

System Identification

- We apply "data science" to building models for dynamical systems from observed input-output signals. (Often for intended model simulation and control design).
- Well established research area within Automatic Control. Term coined in 1957.

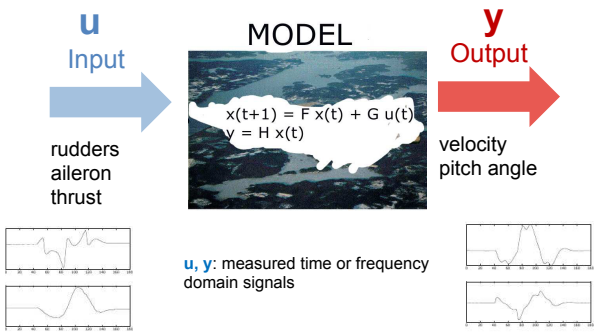
An Introductory Example: System

The System



An Introductory Example 2: Model

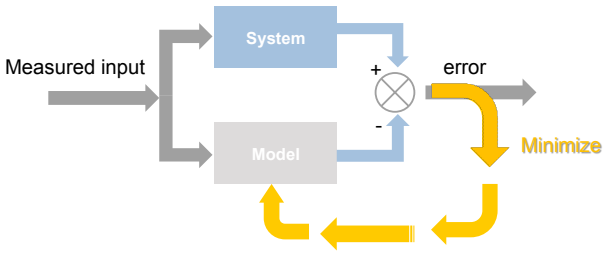
The Model



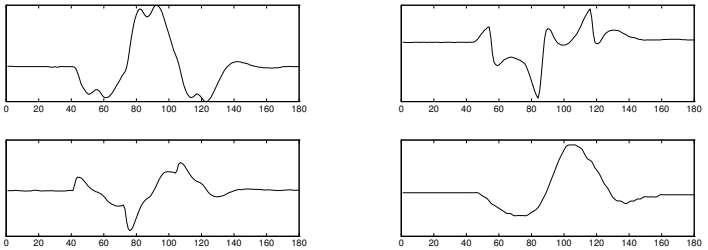
u, y: measured time or frequency domain signals

An Introductory Example 3: Model Fitting

The System and the Model



Data from the Gripen Aircraft



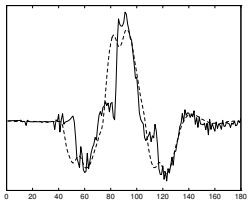
Pitch rate, Canard,
Elevator, Leading Edge Flap

- How do the control surface angles affect the pitch rate?

Aircraft Dynamics: From input 1

$y(t)$ pitch rate at time t. $u_1(t)$ canard angle at time t. $T = 1/60$.
Try

$$y(t) = +b_1u_1(t-T) + b_2u_1(t-2T) + b_3u_1(t-3T) + b_4u_1(t-4T)$$

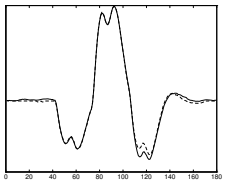


Dashed line: Actual Pitch rate. Solid line: 10 step ahead predicted pitch rate, based on the fourth order model from canard angle only.
First half estimation data - second half validation data.

Using All Inputs

u_1 canard angle; u_2 Elevator angle; u_3 Leading edge flap;

$$\begin{aligned} y(t) = & -a_1y(t-T) - a_2y(t-2T) - a_3y(t-3T) - a_4y(t-4T) \\ & + b_1^1u_1(t-T) + \ldots + b_1^4u_1(t-4T) \\ & + b_2^1u_2(t-T) + \ldots + b_1^3u_3(t-T) + \ldots + b_4^3u_3(t-4T) \end{aligned}$$

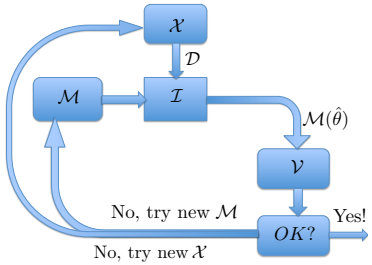


Dashed line: Actual Pitch rate. Solid line: 10 step ahead predicted pitch rate, based on the fourth order model from all inputs.
First half estimation data - second half validation data.

System Identification: Issues

- Select a class of candidate models
- Select a member in this class using the observed data
- Evaluate the quality of the obtained model
- Design the experiment so that the model will be “good”.

The System Identification Flow



$\mathcal{X}$: The Experiment
$\mathcal{D}$: The Measured Data
$\mathcal{M}$: The Model Set
$\mathcal{I}$: The Identification Method
$\mathcal{V}$: The Validation Procedure

Models: General Aspects for Dynamical Systems

- A model is a mathematical expression that describes the connections between measured inputs and outputs, and possibly related noise sequences.
- They can come in many different forms
- The models are labeled with a parameter vector θ
- A common framework is to describe the model as a predictor of the next output, based on observations of past input-output data.
  Observed input–output (u, y) data up to time t: Z^t
  Model described by predictor: $\mathcal{M}(\theta) : \hat{y}(t|\theta) = g(t, \theta, Z^{t-1})$.

Estimation

If a model, $\hat{y}(t|\theta)$, essentially is a predictor of the next output, is is natural to evaluate its quality by assessing how well it predicts: Form the *Prediction error* and measure its size:

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t|\theta), \quad \ell(\varepsilon(t, \theta)) = \varepsilon^2(t, \theta)$$

How has it performed historically?

$$V_N(\theta) = \sum_{t=1}^{N} \ell(\varepsilon(t, \theta))$$

Which model in the structure performed best?

$$\hat{\theta}_N = \arg\min_{\theta \in D_{\mathcal{M}}} V_N(\theta)$$

Often coincides with the Maximum Likelihood Estimate.

Current Issues in System Identification

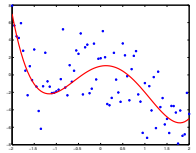
- Full understanding of possibilities and choices of nonlinear model structures
- Best tuning and choice of Regularization:
  $\arg\min \sum_{t=1}^{N} \ell(\varepsilon(t, \theta)) + \theta^T R(\eta)\theta$
- Convexification of estimation formulation (Avoid local minima).
- Rapprochement with machine learning techniques

Connections to Machine Learning

Machine Learning is in the end, I would say, about building a function

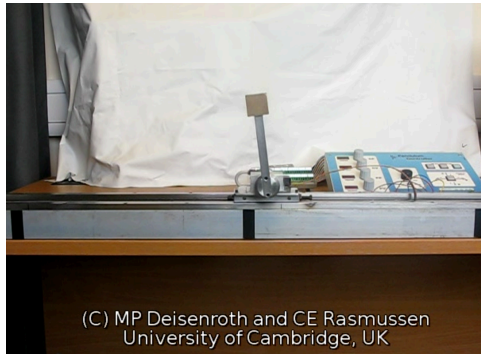
$$y = f(x)$$

from observations of stimuli (regressors) x_t and corresponding (noisy) responses y_t. This is basically a nonlinear regression problem. The data may come from anywhere. The spaces where x and y live can be quite diverse (from high-school curve fitting,



to classification of Facebook texts)

Example: Learning to Stabilize an Inverted Pendulum



100 seconds video: To think about: In this machine learning exercise: what function is estimated?

Variable and Functions

In the experiments, the (state) variables

$$x(t) = \begin{bmatrix} \text{pendulum angle} \\ \text{angular velocity} \\ \text{cart position} \\ \text{cart velocity} \end{bmatrix}$$

and the (input) $u(t)$ = cart acceleration (applied force) were measured, and the (state transition) function f:

$$x(t+1) = f(x(t), u(t)) \qquad (\mathcal{R}^5 \to \mathcal{R}^4)$$

was estimated. The control was decided using a LQG controller for each portion of the experiment based on the current f-estimate.

Model Set: Two ways of parametrizing the function f

Model Structure for f

It is essential for machine learning the the estimated function f is capable of describing **any** function. Two dominating techniques are

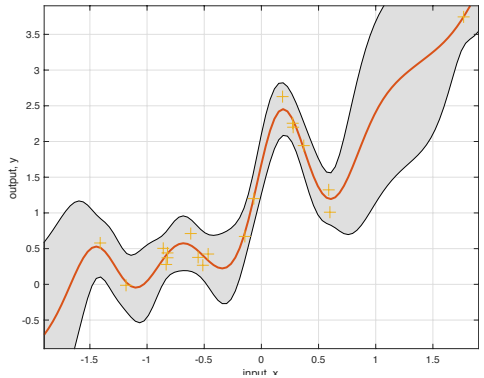
- Gaussian Processes
- Neural Networks

Gaussian Processes

The function to be estimated is embedded in a stochastic framework so that $f(x)$ is seen as a realization of a (Gaussian) stochastic process.
So there is a *prior* (before any measurements) distribution with a (zero) mean, (large) variance and covariance function that describes the smoothness of f.
When $y_k = f(x_k),\ k = 1, \ldots N$ have been measured, the *posterior* distribution $f^p(x|y)$ can be formed for any x. The mean of that function will be the estimate of the function f and is formed by interpolation and extrapolation $[y_k, x_k]$ using the probabilistic relationships. The reliability of the estimate can also be assessed from the posterior variance. [But is of course a reflection of the prior assigned distributions.]
If all variables are jointly Gaussian all this can be computed by simple and efficient linear algebraic expressions.

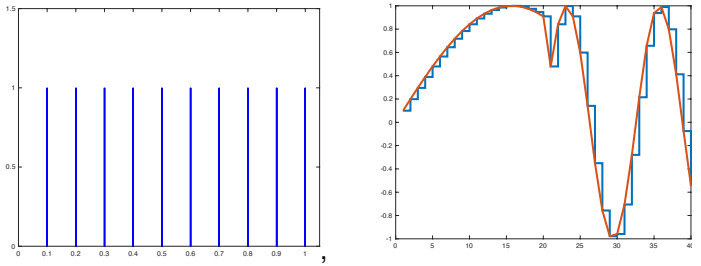
Example

Yellow crosses: the observed x and y values
Red line: The estimate of the f function
Shaded region: Reliability: the standard deviation around the mean f.



Grid-based Techniques (Example for scalar y, x)

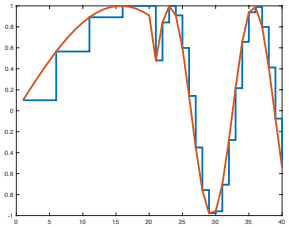
Simple idea: Describe $f(x)$ as piecewise constant over a fine grid of $x, x_k = k * \beta$.



Intuitively clear that this can approximate any reasonable function $f(x)$ with arbitrary accuracy for small enough β.
This corresponds to choosing the grid points as $k\beta, k = 1, \ldots, d$.

Adaptive Grids (Neural Networks)

If the details of $f(x)$ vary with x it is more natural to choose an *adaptive grid*: let the grid points be γ_k with widths β_k and estimate $\gamma_k, \beta_k, k = 1, \ldots, d$ from the data.



This adaptive grid is the essence of an *artificial neural network, ANN*
Mathematically: Let $\kappa(x)$ be the unit pulse (1 for $0 \leq x < 1$, else zero), shift and scale it and use

$$f(x) = \sum_{k=1}^{d} \alpha_k \kappa(\beta_k(x - \gamma_k))$$

Often κ is chosen as the Gaussian Bell (radial basis ANN) or a Sigmoid function (sigmoidal ANN)

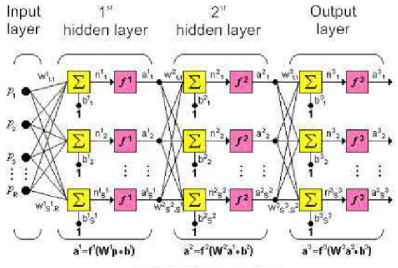
Some Comments on Machine Learning with ANN

$$f(x|\theta) = \sum_{k=1}^{d} \alpha_k \kappa(\beta_k(x - \gamma_k)); \ \theta = [\alpha_k, \beta_k, \gamma_k]$$

- When x is a vector (multiple inputs), β_k are also vectors so $\beta_k x$ is a scalar.
- The estimation of the function f from data y_t, x_t follows the general estimation paradigm: $\min_\theta \sum \|y_t - f(x_t|\theta)\|^2$
- The terms $\kappa(\beta_k(x - \gamma_k))$ are called *hidden units* (since their values do not appear explicitly)
- The collection of the d hidden units is referred to as a *hidden layer*

More Hidden Layers

The outputs of the d hidden units, the d-vector $x^{(2)}(x,\theta) = [\kappa(\beta_k(x-\gamma_k)),\ k = 1,...,d]$ can be seen as a regression vector of processed variables and can be fed into a *second hidden layer*



$$f(x|\theta) = \sum_{k=1}^{d^{(2)}} \alpha_k^{(2)} \kappa(\beta_k^{(2)}(x^{(2)}(x,\theta)) - \gamma_k^{(2)}));$$

Deep Networks

The values of the $d^{(2)}$ hidden units of second layer can in turn be seen as further processed regression variables and be used in a third hidden layer. Etc. So networks with several (many) hidden layers can be created using the same formula as in the first layer. So networks of arbitrary depth can be created, characterized by the number of hidden units in each layer.

Experience has shown that ANNs with the same number of hidden units but more hidden layers show better performance in capturing complicated functions.
Hence the interest in *Deep Neural Networks*, sometimes with hundreds of hidden layers.
How come?

What is the Secret of Deep Networks?

- No clear explanation has been given.
- Possible insight: In pattern recognition it is well known to be useful to base the classification $f(x)$ on selected feature vectors: One way is to select *features* $\Phi_k(x) \in R^p$ by some *projections*. The selection may itself be parameterized: $\Phi_k(x,\eta)$. f will now be a function of $\Phi(x,\eta))$ parameterized by $\theta; f(\theta, \Phi(x,\eta))$.
- Deep networks allow the formation of such features from which the model can be constructed.

Deep Learning

Applying the learning algorithm (minimizing the fit

$$\min_\theta \sum \|y_t - f(x_t|\theta)\|^2$$

for a deep network f) is known as *Deep Learning* and has been applied in the last years to difficult problems with surprising and spectacular success.
Deep learning with the complicated networks requires extensive calculations and a massive amount of data

Big Data

- Big Data is a hyped term, applied in many problem areas.
- It basically refers to handling the massive amount of data that are coming available. IoT Internet of things, (“K,M,G,T,P,Z”)
- A basic challenge is the handling of the data – Database technologies need to be further developed.
- Here we shall briefly comment on big data opportunities for the process industry.

Process Industrial Databases

PM 12, Stora Enso, Borlange, Sweden: 75000 control signals, 15000 control loops. All process variables sampled at 1Hz for 100 years: 0.1 PetaByte of data.

Data mining in large historic process databases.

- The database is the “ultimate process model”
- But the data need to be indexed and marked with quality and relevance markings.
- Running a recursive algoritm along the data, marking information matrix values

Predictive Maintenance

Another important opportunity for big data is predictive maintenance: Monitor important process variables and estimate e.g. RUL: Remaining Useful Lifetime and alert for time for maintenance and imminent failures.

- MathWorks released in March the PREDICTIVE MAINTENANCE TOOLBOX for MATLAB.
- ABB offers “ABB Ability^{TM}” for manufacturing when they sell Industrial Robots

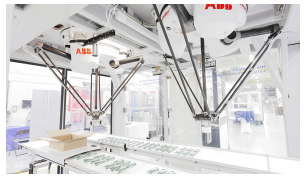


ABB Ability^{TM} for Manufacturing

The ABB initiative is to offer the customers to on-line send robot data measurements via the ABB cloud to a distributed database (potentially covering $\sim 10^5$ robots).
The data is analysed - and the customers are alerted to imminent maintenance needs for their robot(s). At the same time knowledge is gathered at ABB for the performance and properties of a large fleet of robots – providing continuously improved basis for the decisions.
The data are subjected to the most modern statistical tools for classification and pattern recognition, (logistic regression, random forests, extreme gradient boosting, ANNs, etc)
This shows, that even for big data, the analysis, in the end relies upon tools from established theory.
[BTW: The Predictive Maintenance Toolbox requires the System Identification Toolbox as a prerequisite.]

Conclusions

- There are a few central concepts for all data science.
- “Classical System Identification” serves as a fine template for most of modern data science.
- Machine learning, deep learning and part of big data clearly relate to central system identification issues.