Combining kernel-based methods and the EM method for structured system identification

Giulio Bottegal

KTH - Royal Institute of Technology (Sweden)

September 21, 2015

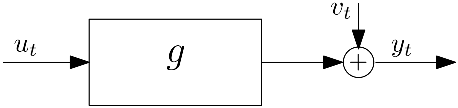
Outline

- Kernel-based linear system identification
- Handling input uncertainties
- Handling outliers

Outline

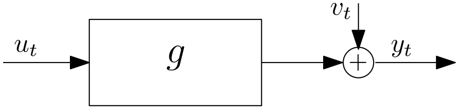
- Kernel-based linear system identification
- Handling input uncertainties
- Handling outliers

A standard system identification problem



Model in time domain: $y_t = (g * u)_t + v_t$

A standard system identification problem



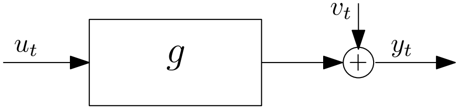
Model in time domain: $y_t = (g * u)_t + v_t$

Assumptions

- g BIBO stable, unknown order
- v_t white Gaussian (unknown variance σ^2)

Goal: Estimate $g_1, \ldots, g_n$ (n large enough) from $\{u_t, y_t\}_{t=1}^N$

A standard system identification problem



Model in time domain: $y_t = (g * u)_t + v_t$

Assumptions

- g BIBO stable, unknown order
- v_t white Gaussian (unknown variance σ^2)

Goal: Estimate $g_1, \ldots, g_n$ (n large enough) from $\{u_t, y_t\}_{t=1}^N$

Linear regression problem: $y = Ug + v$

A Bayesian point of view

g **is a Gaussian vector:** $g \sim \mathcal{N}(0, \lambda K_\beta)$

A Bayesian point of view

g is a Gaussian vector: $g \sim \mathcal{N}(0, \lambda K_\beta)$

Stable spline kernel

[Pillonetto & De Nicolao,'10],[Chen, Ohlsson, Ljung,'12]

$$K_\beta(i,j) := \beta^{\max(i,j)} \qquad 0 < \beta < 1$$

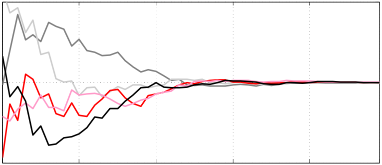
A Bayesian point of view

g **is a Gaussian vector:** $g \sim \mathcal{N}(0,\, \lambda K_\beta)$

Stable spline kernel

[Pillonetto & De Nicolao,'10],[Chen, Ohlsson, Ljung,'12]

$$K_\beta(i,j) := \beta^{\max(i,j)} \qquad 0 < \beta < 1$$



BIBO stable realizations

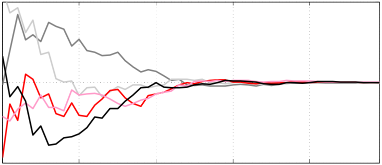
A Bayesian point of view

g **is a Gaussian vector:** $g \sim \mathcal{N}(0, \lambda K_\beta)$

Stable spline kernel

[Pillonetto & De Nicolao,'10],[Chen, Ohlsson, Ljung,'12]

$$K_\beta(i,j) := \beta^{\max(i,j)} \qquad 0 < \beta < 1$$



BIBO stable realizations

Two hyperparameters to be chosen

1. λ: amplitude of the impulse response (> 0)
2. β: related to the decay rate of g

Estimation of the impulse response

Exploiting joint Gaussianity between g and y...

$$p(g|y) = \mathcal{N}(Cy,\, P)$$

$$P = \left(\frac{U^T U}{\sigma^2} + (\lambda K_\beta)^{-1} \right)^{-1} \qquad C = P \frac{U^T}{\sigma^2}$$

Estimation of the impulse response

Exploiting joint Gaussianity between g and y...

$$p(g|y) = \mathcal{N}(Cy,\, P)$$

$$P = \left(\frac{U^T U}{\sigma^2} + (\lambda K_\beta)^{-1} \right)^{-1} \qquad C = P \frac{U^T}{\sigma^2}$$

MMSE estimator of g:

$$\hat{g} = \mathbb{E}[g|y] = C(\theta) y$$

Estimation of the impulse response

Exploiting joint Gaussianity between g and y...

$$p(g|y) = \mathcal{N}(Cy,\ P)$$

$$P = \left(\frac{U^T U}{\sigma^2} + (\lambda K_\beta)^{-1} \right)^{-1} \qquad C = P \frac{U^T}{\sigma^2}$$

MMSE estimator of g:

$$\hat{g} = \mathbb{E}[g|y] = C(\theta) y$$

... but it depends on the parameter vector $\theta = [\lambda,\ \beta,\ \sigma^2]$

Parameter selection

Empirical Bayes approach *(Pillonetto, Chiuso, De Nicolao, '11)*

- Marginal distribution of y:

$$p(y) = \int p(y, g) dg$$

Parameter selection

Empirical Bayes approach *(Pillonetto,Chiuso,De Nicolao, '11)*

- Marginal distribution of y:

$$p(y) = \int p(y,\, g)dg$$

- Perform marginal likelihood optimization

$$\hat{\theta} = \arg\max \log p(y;\, \theta)$$

Parameter selection

Empirical Bayes approach *(Pillonetto, Chiuso, De Nicolao, '11)*

- Marginal distribution of y:

$$p(y) = \int p(y,\, g) dg$$

- Perform marginal likelihood optimization

$$\hat{\theta} = \arg\max \log p(y;\, \theta)$$

- Simple optimization problem $\longrightarrow$ Matlab's `fminsearch`

Parameter selection

Empirical Bayes approach *(Pillonetto, Chiuso, De Nicolao, '11)*

- Marginal distribution of y:

$$p(y) = \int p(y,\, g)dg$$

- Perform marginal likelihood optimization

$$\hat{\theta} = \arg\max \log p(y;\, \theta)$$

- Simple optimization problem $\longrightarrow$ Matlab's `fminsearch`

Can we come up with other optimization strategies?

Optimizing via EM

The EM method

1. Define "missing data": our unknown system g
2. Introduce the complete likelihood: $L(y, g; \theta) := \log p(y, g; \theta)$

Optimizing via EM

The EM method

1. Define "missing data": our unknown system g
2. Introduce the complete likelihood: $L(y, g; \theta) := \log p(y, g; \theta)$

Iterations

- $\hat{\theta}^{(k)} :=$ estimate of θ at $k-$th iteration
- for every k, until convergence, do:

Optimizing via EM

The EM method

1. Define "missing data": our unknown system g
2. Introduce the complete likelihood: $L(y, g; \theta) := \log p(y, g; \theta)$

Iterations

- $\hat{\theta}^{(k)} :=$ estimate of θ at $k-$th iteration
- for every k, until convergence, do:
    1. E-step: Compute $Q(\theta, \hat{\theta}^{(k)}) := \mathbb{E}_{p(g|y, \hat{\theta}^{(k)})}[L(y, g|\theta)]$

Optimizing via EM

The EM method

1. Define "missing data": our unknown system g
2. Introduce the complete likelihood: $L(y, g; \theta) := \log p(y, g; \theta)$

Iterations

- $\hat{\theta}^{(k)} :=$ estimate of θ at $k-$th iteration
- for every k, until convergence, do:
  1. E-step: Compute $Q(\theta, \hat{\theta}^{(k)}) := \mathbb{E}_{p(g|y,\, \hat{\theta}^{(k)})}[L(y, g|\theta)]$
  2. M-step: Solve $\hat{\theta}^{(k+1)} := \arg\max Q(\theta, \hat{\theta}^{(k)})$

Optimizing via EM

The EM method

1. Define "missing data": our unknown system g
2. Introduce the complete likelihood: $L(y,\, g;\, \theta) := \log p(y,\, g;\, \theta)$

Iterations

- $\hat{\theta}^{(k)} :=$ estimate of θ at $k-$th iteration
- for every k, until convergence, do:
  1. E-step: Compute $Q(\theta, \hat{\theta}^{(k)}) := \mathbb{E}_{p(g|y,\, \hat{\theta}^{(k)})}\left[L(y,\, g|\theta)\right]$
  2. M-step: Solve $\hat{\theta}^{(k+1)} := \arg\max Q(\theta, \hat{\theta}^{(k)})$

Convergence properties

[Dempster et al., 1977]: $\hat{\theta}^{(k)} \longrightarrow$ local solution of $\arg\max \log p(y;\theta)$

EM in our problem

At each iteration, compute

$$\hat{g}^{(k)} = C(\theta^{(k)})y \qquad P^{(k)} = \mathrm{Cov}\left[g^{(k)}\right]$$

Recall: $\theta = [\lambda,\, \beta,\, \sigma^2]$

EM in our problem

At each iteration, compute

$$\hat{g}^{(k)} = C(\theta^{(k)})y \qquad P^{(k)} = \mathrm{Cov}\left[g^{(k)}\right]$$

Recall: $\theta = [\lambda,\, \beta,\, \sigma^2]$

Parameter update

- $\hat{\lambda}^{(k+1)} = \frac{1}{n}\mathrm{tr}\left[K_{\hat{\beta}^{(k)}}^{-1}(P^{(k)} + \hat{g}^{(k)}\hat{g}^{(k)T})\right]$ $\hookleftarrow$ closed-form

EM in our problem

At each iteration, compute

$$\hat{g}^{(k)} = C(\theta^{(k)})y \qquad P^{(k)} = \mathrm{Cov}\left[g^{(k)}\right]$$

Recall: $\theta = [\lambda,\, \beta,\, \sigma^2]$

Parameter update

- $\hat{\lambda}^{(k+1)} = \frac{1}{n}\mathrm{tr}\left[K^{-1}_{\hat{\beta}^{(k)}}(P^{(k)} + \hat{g}^{(k)}\hat{g}^{(k)T})\right]$ $\hookleftarrow$ closed-form
- $\hat{\beta}^{(k+1)} = \arg\max Q(\beta,\, \hat{\theta}^{(k)})$ $\hookleftarrow$ scalar problem

EM in our problem

At each iteration, compute

$$\hat{g}^{(k)} = C(\theta^{(k)})y \qquad P^{(k)} = \mathrm{Cov}\left[g^{(k)}\right]$$

Recall: $\theta = [\lambda,\ \beta,\ \sigma^2]$

Parameter update

- $\hat{\lambda}^{(k+1)} = \frac{1}{n}\mathrm{tr}\left[K^{-1}_{\hat{\beta}^{(k)}}(P^{(k)} + \hat{g}^{(k)}\hat{g}^{(k)T})\right]$ ↩ closed-form
- $\hat{\beta}^{(k+1)} = \arg\max Q(\beta,\ \hat{\theta}^{(k)})$ ↩ scalar problem
- $\hat{\sigma}^{2,(k+1)} = \frac{1}{N}\left(\|y - U\hat{g}^{(k)}\|^2 + \mathrm{tr}\left\{UP^{(k)}U^T\right\}\right)$ ↩ closed-form

EM in our problem

At each iteration, compute

$$\hat{g}^{(k)} = C(\theta^{(k)})y \qquad P^{(k)} = \mathrm{Cov}\left[g^{(k)}\right]$$

Recall: $\theta = [\lambda,\, \beta,\, \sigma^2]$

Parameter update

- $\hat{\lambda}^{(k+1)} = \frac{1}{n}\mathrm{tr}\left[K^{-1}_{\hat{\beta}^{(k)}}(P^{(k)} + \hat{g}^{(k)}\hat{g}^{(k)T})\right]$ ⟵ closed-form
- $\hat{\beta}^{(k+1)} = \arg\max Q(\beta,\, \hat{\theta}^{(k)})$ ⟵ scalar problem
- $\hat{\sigma}^{2,(k+1)} = \frac{1}{N}\left(\|y - U\hat{g}^{(k)}\|^2 + \mathrm{tr}\left\{UP^{(k)}U^T\right\}\right)$ ⟵ closed-form

Simple updates... but is it really worth it?

Outline

- Kernel-based linear system identification
- Handling input uncertainties
- Handling outliers

Input uncertainties

Assumption

$$u = \begin{bmatrix} u_1 & \dots & u_N \end{bmatrix}^T \quad \longrightarrow u = Hx \quad , \quad x \in \mathbb{R}^p$$

H known matrix, x unknown vector $\hookleftarrow$ needs to be estimated

Input uncertainties

Assumption

$$u = \begin{bmatrix} u_1 & \dots & u_N \end{bmatrix}^T \quad \longrightarrow u = Hx \quad , \quad x \in \mathbb{R}^p$$

H known matrix, x unknown vector $\hookleftarrow$ needs to be estimated

How to estimate x?

Include x in the ML optimization: $\theta := [x^T \ \lambda \ \beta \ \sigma^2]$

$$\hat{\theta} = \arg\max \log p(y;\, \theta)$$

Input uncertainties

Assumption

$$u = \begin{bmatrix} u_1 & \dots & u_N \end{bmatrix}^T \quad \longrightarrow u = Hx \quad , \quad x \in \mathbb{R}^p$$

H known matrix, x unknown vector ← needs to be estimated

How to estimate x?

Include x in the ML optimization: $\theta := [x^T \ \lambda \ \beta \ \sigma^2]$

$$\hat{\theta} = \arg\max \log p(y;\, \theta)$$

EM estimation

Given the k-th iteration parameter guess $\hat{\theta}^{(k)}$, compute

- $\hat{x}^{(k+1)} = (A^{(k)})^{-1} b^{(k)}$ ← $A^{(k)}$ and $b^{(k)}$ easily computed
- $\hat{\lambda}^{(k+1)}$, $\hat{\beta}^{(k+1)}$, $\hat{\sigma}^{2,(k+1)}$ as before ← use $\hat{u}^{(k)} = H\hat{x}^{(k)}$

Input uncertainties (2)

- Input parameters enter EM iterations smoothly

Input uncertainties (2)

- Input parameters enter EM iterations smoothly
- ... but in which scenarios does this problem pop up?

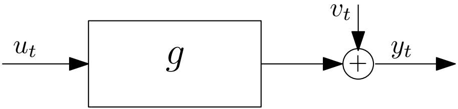
Input uncertainties (2)

- Input parameters enter EM iterations smoothly
- ... but in which scenarios does this problem pop up?

Examples

- Semi-blind system identification
- Hammerstein system identification
- Estimation of initial conditions

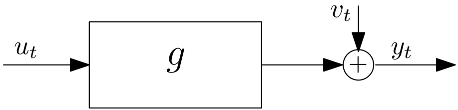
Semi-blind system identification



- Same setup as before...
- but no measurements of u_t

G. Bottegal, R.S. Risuleo, H. Hjalmarsson.
Blind system identification using kernel-based methods. *IFAC SysId 2015*

Semi-blind system identification



- Same setup as before...
- but no measurements of u_t

Assumption

u belongs to a p dimensional subspace (semi-blind)

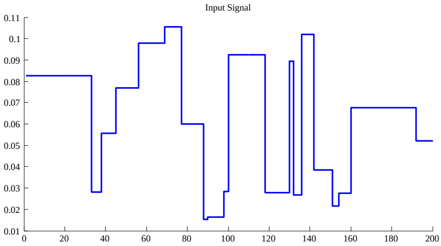
$\hookrightarrow$ We know H such that $u = Hx$

G. Bottegal, R.S. Risuleo, H. Hjalmarsson.

Blind system identification using kernel-based methods. *IFAC SysId 2015*

Input models

Example: u piecewise constant with known switching instants

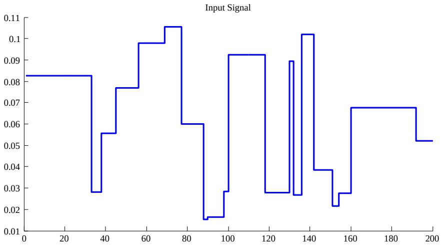


$$H = \begin{bmatrix} \mathbf{1}_{T_1} & & & \\ & \mathbf{1}_{T_2 - T_1} & & \\ & & \ddots & \\ & & & \mathbf{1}_{T_p - T_{p-1}} \end{bmatrix}$$

$x =$ input levels

Input models

Example: u piecewise constant with known switching instants



$$H = \begin{bmatrix} \mathbf{1}_{T_1} & & & \\ & \mathbf{1}_{T_2-T_1} & & \\ & & \ddots & \\ & & & \mathbf{1}_{T_p-T_{p-1}} \end{bmatrix}$$

$x =$ input levels

Applications:

- Occupancy estimation from CO_2 measurements

  (A. Ebadat et al. *Blind identification strategies for room occupancy estimation.* ECC '15)

- Non-intrusive load monitoring

Identification procedure

Identifiability

Any pair $\left(g,\, \frac{1}{\alpha}u\right)$ explains the data equally well!

$\hookrightarrow$ Assume $\|g\|_2 = 1$ and $g_1 > 0$

Identification procedure

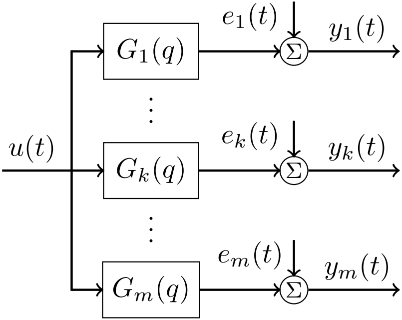
Identifiability

Any pair $\left(g,\ \frac{1}{\alpha}u\right)$ explains the data equally well!
$\hookrightarrow$ Assume $\|g\|_2 = 1$ and $g_1 > 0$

Identification

- Fix $\lambda = 1$ and repeat EM steps until convergence:
  1. update $\hat{x}$
  2. update $\hat{\sigma}^2$
  3. update $\hat{\beta}$
- Compute $\hat{u} = H\hat{x}$
- Compute $\hat{g} = C(\hat{\theta})y$ and normalize

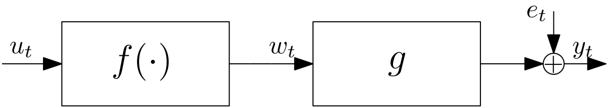
Full blind system identification



SIMO system

- *Enough* subsystems?
  $\hookrightarrow$ Full blind system identification ($H = I$)
- Noise spatial correlation can be included

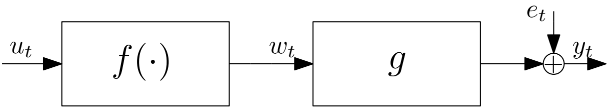
Hammerstein system identification



$$w_t = f(u_t)$$
$$y_t = \sum_{k=1}^{\infty} g_k w_{t-k} + e_t$$

R.S. Risuleo, G. Bottegal, H. Hjalmarsson.
A kernel-based approach to Hammerstein system identification. *IFAC SysId 2015*

Hammerstein system identification



$$w_t = f(u_t)$$
$$y_t = \sum_{k=1}^{\infty} g_k w_{t-k} + e_t$$

Modeling approach

- p basis functions for f: $f(u_t) = \sum_{i=1}^{p} x_i h_i(u_t)$ $\hookleftarrow$ only h_i's known
- kernel-based model of g, with $\|g\|_2 = 1, g_1 > 0$ $\hookleftarrow$ for identifiability

---

R.S. Risuleo, G. Bottegal, H. Hjalmarsson.

A kernel-based approach to Hammerstein system identification. *IFAC SysId 2015*

Identification with EM

How to cast Hammerstein system identification into our framework?

$$w = \begin{bmatrix} \sum x_i h_i(u_1) \\ \vdots \\ \sum x_i h_i(u_N) \end{bmatrix} = Hx$$

Identification with EM

How to cast Hammerstein system identification into our framework?

$$w = \begin{bmatrix} \sum x_i h_i(u_1) \\ \vdots \\ \sum x_i h_i(u_N) \end{bmatrix} = Hx$$

- Identification: same algorithm as semi-blind system identification

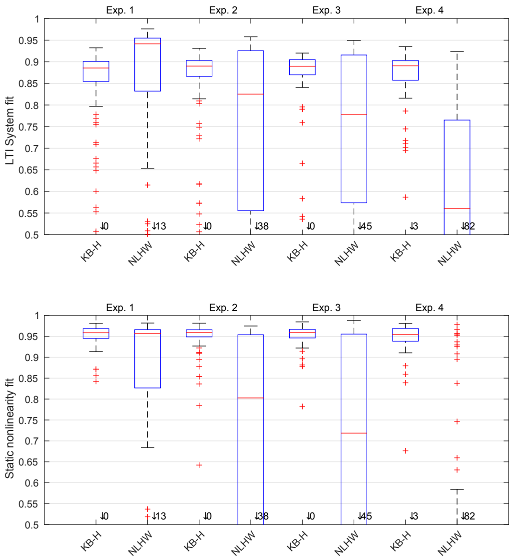
Identification with EM

How to cast Hammerstein system identification into our framework?

$$w = \begin{bmatrix} \sum x_i h_i(u_1) \\ \vdots \\ \sum x_i h_i(u_N) \end{bmatrix} = Hx$$

- Identification: same algorithm as semi-blind system identification
- Extension: nonparametric models of f (see next poster session)

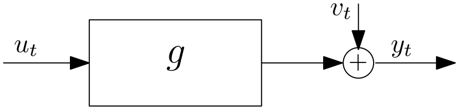
Simulations



Features

- $N = 500$
- $n = 100$
- SNR $= 10$
- LTI orders $= [4,8,10,20]$
- 7th order polynomial

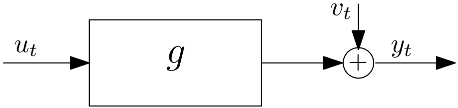
Estimating the initial conditions



Goal: Estimate $g_1, \ldots, g_n$ from $\{u_t, y_t\}_{t=0}^{N}$

R.S. Risuleo, G. Bottegal, H. Hjalmarsson.
On the estimation of initial conditions in kernel-based system identification.
IEEE CDC 2015

Estimating the initial conditions



Goal: Estimate $g_1, \ldots, g_n$ from $\{u_t, y_t\}_{t=0}^{N}$

Recall: $y = Ug + v$

R.S. Risuleo, G. Bottegal, H. Hjalmarsson.
On the estimation of initial conditions in kernel-based system identification.
IEEE CDC 2015

Initial conditions

The matrix U depends on unavailable input samples!

$$U := \begin{bmatrix} u_0 & u_{-1} & u_{-2} & & u_{-n+1} \\ u_1 & u_0 & u_{-1} & & u_{-n+2} \\ u_2 & u_1 & u_0 & \cdots & u_{-n+3} \\ \vdots & \vdots & \vdots & & \vdots \\ u_{N-1} & u_{N-2} & u_{N-3} & & u_{N-n} \end{bmatrix}$$

Solutions?

How do we cope with initial conditions?

Initial conditions

The matrix U depends on unavailable input samples!

$$U := \begin{bmatrix} & & & & \\ & & & & \\ u_{n-1} & u_{n-2} & u_{n-3} & \cdots & u_0 \\ \vdots & \vdots & \vdots & & \vdots \\ u_{N-1} & u_{N-2} & u_{N-3} & \cdots & u_{N-n} \end{bmatrix}$$

Solutions?

Truncation $\hookleftarrow$ throw away n measurements!

Initial conditions

The matrix U depends on unavailable input samples!

$$U := \begin{bmatrix} u_0 & 0 & 0 & & 0 \\ u_1 & u_0 & 0 & & 0 \\ u_2 & u_1 & u_0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ u_{N-1} & u_{N-2} & u_{N-3} & & u_{N-n} \end{bmatrix}$$

Solutions?

Set to 0 $\hookleftarrow$ may be very inaccurate...

Initial conditions

The matrix U depends on unavailable input samples!

$$U := \begin{bmatrix} u_0 & \hat{u}_{-1} & \hat{u}_{-2} & & \hat{u}_{-n+1} \\ u_1 & u_0 & \hat{u}_{-1} & & \hat{u}_{-n+2} \\ u_2 & u_1 & u_0 & \cdots & \hat{u}_{-n+3} \\ \vdots & \vdots & \vdots & & \vdots \\ u_{N-1} & u_{N-2} & u_{N-3} & & u_{N-n} \end{bmatrix}$$

Solutions?

Estimate them from data!

EM estimation

C.I. estimation + system identification

- Just set $x = [u_{-1} \; \ldots \; u_{-n+1}]$ $\qquad (H = I)$
- Apply EM-based identification procedure!

EM estimation

C.I. estimation + system identification

- Just set $x = [u_{-1} \; \dots \; u_{-n+1}]$ $\qquad (H = I)$
- Apply EM-based identification procedure!

N	150	250	400
Truncation	42.01	59.40	62.96
Zeros	51.69	61.15	63.18
EM-based	55.69	62.77	64.45
Oracle	57.31	63.78	64.95

Features

$n = 100$, $SNR = 20$, $u =$ ARMA process, system order $= 40$

Outline

- Kernel-based linear system identification
- Handling input uncertainties
- Handling outliers

Outliers in system identification

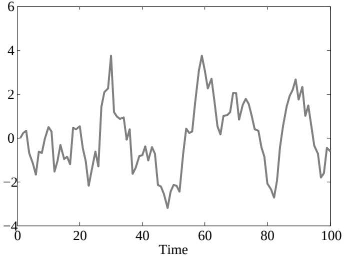
Why should we contrast outliers?

G. Bottegal, A.Y Aravkin, H. Hjalmarsson, G. Pillonetto.
Robust EM kernel-based methods for linear system identification.
Automatica (provisionally accepted)

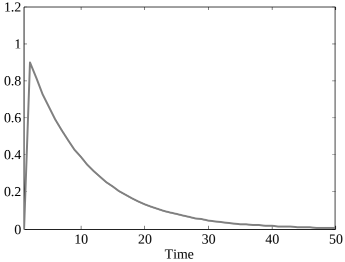
Outliers in system identification

Why should we contrast outliers?

Noiseless output



True impulse response

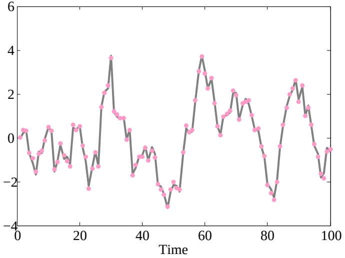


G. Bottegal, A.Y Aravkin, H. Hjalmarsson, G. Pillonetto.
Robust EM kernel-based methods for linear system identification.
Automatica (provisionally accepted)

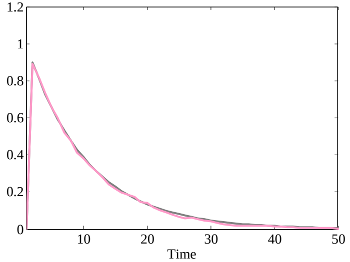
Outliers in system identification

Why should we contrast outliers?

Noisy output



Estimated impulse response

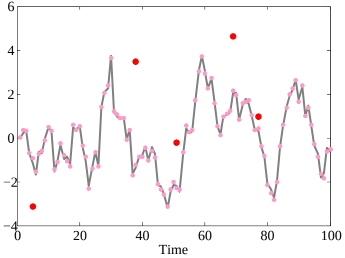


G. Bottegal, A.Y Aravkin, H. Hjalmarsson, G. Pillonetto.
Robust EM kernel-based methods for linear system identification.
Automatica (provisionally accepted)

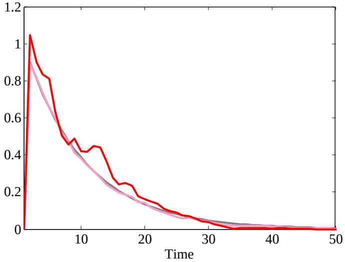
Outliers in system identification

Why should we contrast outliers?

Outliers

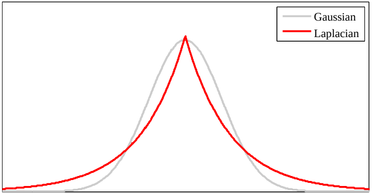


Estimated impulse response



G. Bottegal, A.Y Aravkin, H. Hjalmarsson, G. Pillonetto.
Robust EM kernel-based methods for linear system identification.
Automatica (provisionally accepted)

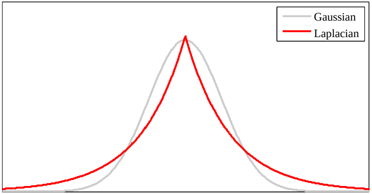
How to model outliers?

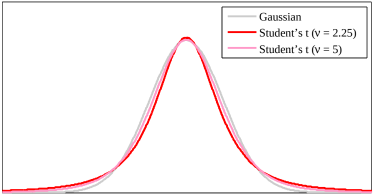


Long tailed random variables

- Laplacian density: $p(v_t) = \frac{1}{\sqrt{2}\sigma} e^{-\frac{\sqrt{2}|v_t|}{\sigma}}$

How to model outliers?



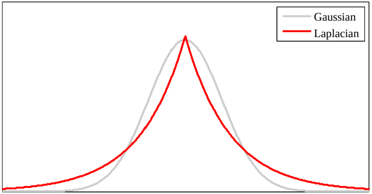


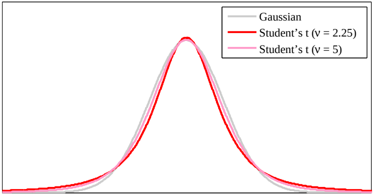
Long tailed random variables

- Laplacian density: $p(v_t) = \frac{1}{\sqrt{2}\sigma} e^{-\frac{\sqrt{2}|v_t|}{\sigma}}$
- Student's t density: $p(v_t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi(\nu-2)\sigma^2}} \left(1 + \frac{v_t^2}{(\nu-2)\sigma^2}\right)^{-\frac{\nu+1}{2}}$

  ($\nu :=$ degrees of freedom)

How to model outliers?





Long tailed random variables

- Laplacian density: $p(v_t) = \frac{1}{\sqrt{2}\sigma} e^{-\frac{\sqrt{2}|v_t|}{\sigma}}$
- Student's t density: $p(v_t) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi(\nu-2)\sigma^2}} \left(1 + \frac{v_t^2}{(\nu-2)\sigma^2}\right)^{-\frac{\nu+1}{2}}$

  ($\nu :=$ degrees of freedom)

Long tails $\longrightarrow$ Outliers are expected in this model

Handling non-Gaussianity

v_t non-Gaussian in our model $\longrightarrow$ No closed-form of ML and $\hat{g}$

Handling non-Gaussianity

v_t non-Gaussian in our model $\longrightarrow$ No closed-form of ML and $\hat{g}$

Scale mixture of Gaussians

$$p(v_t) = \int_0^{+\infty} \frac{1}{\sqrt{2\pi\tau_t}} e^{-\frac{v_t^2}{2\tau_t}} \quad \pi(\tau_t) \quad d\tau_t$$

Handling non-Gaussianity

v_t non-Gaussian in our model $\longrightarrow$ No closed-form of ML and $\hat{g}$

Scale mixture of Gaussians

$$p(v_t) = \int_0^{+\infty} \underbrace{\frac{1}{\sqrt{2\pi\tau_t}} e^{-\frac{v_t^2}{2\tau_t}}}_{Gaussian} \underbrace{\pi(\tau_t)}_{Prior\ of\ \tau_t} \, d\tau_t$$

Interpretation: v_t is Gaussian r.v. whose variance τ_t is r.v. with pdf $\pi(\tau_t)$

Handling non-Gaussianity

v_t non-Gaussian in our model $\longrightarrow$ No closed-form of ML and $\hat{g}$

Scale mixture of Gaussians

$$p(v_t) = \int_0^{+\infty} \underbrace{\frac{1}{\sqrt{2\pi\tau_t}} e^{-\frac{v_t^2}{2\tau_t}}}_{Gaussian} \underbrace{\pi(\tau_t)}_{Prior\ of\ \tau_t} \; d\tau_t$$

Interpretation: v_t is Gaussian r.v. whose variance τ_t is r.v. with pdf $\pi(\tau_t)$

What is $\pi(\tau_t)$?

- Laplacian noise $\Longrightarrow$ τ_t Exponential of parameter σ^2
- Student's t noise $\Longrightarrow$ τ_t Inverse Gamma of parameters $\left(\frac{\nu}{2}, \frac{(\nu-2)\sigma^2}{2}\right)$

Handling non-Gaussianity

v_t non-Gaussian in our model $\longrightarrow$ No closed-form of ML and $\hat{g}$

Scale mixture of Gaussians

$$p(v_t) = \int_0^{+\infty} \underbrace{\frac{1}{\sqrt{2\pi\tau_t}} e^{-\frac{v_t^2}{2\tau_t}}}_{Gaussian} \underbrace{\pi(\tau_t)}_{Prior\ of\ \tau_t} \ d\tau_t$$

Interpretation: v_t is Gaussian r.v. whose variance τ_t is r.v. with pdf $\pi(\tau_t)$

What is $\pi(\tau_t)$?

- Laplacian noise $\Longrightarrow$ τ_t Exponential of parameter σ^2
- Student's t noise $\Longrightarrow$ τ_t Inverse Gamma of parameters $\left(\frac{\nu}{2}, \frac{(\nu-2)\sigma^2}{2}\right)$

Need to estimate the new parameters τ_t

MAP estimate of parameters

- $\theta := [\lambda \ \beta \ \tau_1 \ \ldots \ \tau_N]$
- Define $p(\theta)$ prior for θ (given by prior on τ_t)

MAP estimate of parameters

- $\theta := [\lambda \; \beta \; \tau_1 \; \dots \; \tau_N]$
- Define $p(\theta)$ prior for θ (given by prior on τ_t)

MAP: $\hat{\theta} = \arg\max \log \left[p(y|\theta) p(\theta) \right]$

MAP estimate of parameters

- $\theta := [\lambda \ \beta \ \tau_1 \ \ldots \ \tau_N]$
- Define $p(\theta)$ prior for θ (given by prior on τ_t)

MAP: $\hat{\theta} = \arg\max \log \left[p(y|\theta) p(\theta) \right]$

EM solution of MAP problem

- Update $\hat{\lambda}^{(k+1)}$ and $\hat{\beta}^{(k+1)}$ as usual

MAP estimate of parameters

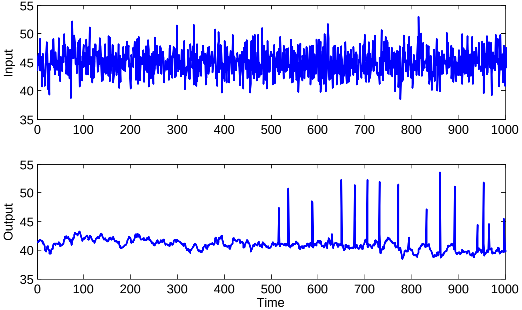
- $\theta := [\lambda \ \beta \ \tau_1 \ \ldots \ \tau_N]$
- Define $p(\theta)$ prior for θ (given by prior on τ_t)

$$\textbf{MAP: } \hat{\theta} = \arg\max \log\left[p(y|\theta)p(\theta)\right]$$

EM solution of MAP problem

- Update $\hat{\lambda}^{(k+1)}$ and $\hat{\beta}^{(k+1)}$ as usual
- Update rule for each τ_t is
  1. Laplacian case: $\hat{\tau}_t^{(k+1)} = \frac{\sigma^2}{4}\left(\sqrt{1+\frac{8\hat{\alpha}_t^{(k)}}{\sigma^2}}-1\right)$
  2. Student's t case: $\hat{\tau}_t^{(k+1)} = \frac{\hat{\alpha}_t^{(k)}+(\nu-2)\sigma^2}{\nu+3}$ ($\hat{\alpha}_t^{(k)}$ *known*)

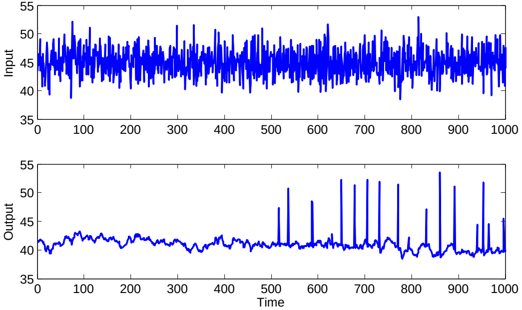
A real experiment



Features

- Input: Voltage to pump
- Output: Water tank level
- Outliers due to pressure perturbation
- Identification using second part of data

A real experiment



Features

- Input: Voltage to pump
- Output: Water tank level
- Outliers due to pressure perturbation
- Identification using second part of data

Method	Fit % on first part of data
Student	67.40
Laplace	51.81
Gaussian	41.49
Gauss. (estimated using test set)	70.06

Conclusions

Starting point: kernel-based linear system identification
$\hookrightarrow$ requires tuning of some (hyper-)parameters

Conclusions

Starting point: kernel-based linear system identification

$\hookrightarrow$ requires tuning of some (hyper-)parameters

Our contribution

- We can select parameters via ML+EM

Conclusions

Starting point: kernel-based linear system identification
$\hookrightarrow$ requires tuning of some (hyper-)parameters

Our contribution

- We can select parameters via ML+EM
- Nice iterative method in case of input uncertainties:
  - Blind system identification
  - Hammerstein systems
  - Estimation of initial conditions

Conclusions

Starting point: kernel-based linear system identification

$\hookrightarrow$ requires tuning of some (hyper-)parameters

Our contribution

- We can select parameters via ML+EM
- Nice iterative method in case of input uncertainties:
  - Blind system identification
  - Hammerstein systems
  - Estimation of initial conditions
- Nice iterative method for outlier robust system identification

Lots of questions!

Open questions

- Combine input uncertainties with robust technique?
- Compare with other iterative gradient-based methods?
- Other challenging problems (Wiener, networks, EIV,...)?
- Convergence rate?
- ...

Lots of questions!

Open questions

- Combine input uncertainties with robust technique?
- Compare with other iterative gradient-based methods?
- Other challenging problems (Wiener, networks, EIV,...)?
- Convergence rate?
- ...

A special thank goes to...

- Riccardo S. Risuleo
- Håkan Hjalmarsson
- Gianluigi Pillonetto

Combining kernel-based methods and the EM method for structured system identification

Giulio Bottegal

KTH - Royal Institute of Technology (Sweden)

September 21, 2015

e-mail: bottegal@kth.se