Outlier-robust estimation of uncertain-input systems with applications to nonparametric FIR and Hammerstein models

Riccardo Sven Risuleo and Håkan Hjalmarsson

KTH - Royal Institute of Technology

hjalmars@kth.se

Keio University, January 18, 2019
Overview

Introduction

Modeling the uncertain-input system

Inference in uncertain-input models

Empirical Bayes estimation algorithm

Examples

Outlier Robustness

Conclusions
Systems with uncertain inputs
Systems with uncertain inputs

\[ w \rightarrow S \rightarrow y \]

\[ \eta \rightarrow v \]

\[ \varepsilon \rightarrow y \]

errors-in-variables

An unknown input signal
Systems with uncertain inputs

\[ S \eta v + \varepsilon + y \] errors-in-variables

\[ S \varepsilon + y \] Hammerstein

\[ S \begin{array}{c} w \\ \eta \end{array} \quad \begin{array}{c} w \\ \varepsilon \end{array} \quad \begin{array}{c} u \\ \varepsilon \end{array} \quad \begin{array}{c} w \\ \varepsilon \end{array} \]
Systems with uncertain inputs

**errors-in-variables**

```
\[ w \rightarrow S \rightarrow + \rightarrow y \]
```

**Hammerstein**

```
\[ u \rightarrow f(\cdot) \rightarrow w \rightarrow S \rightarrow + \rightarrow y \]
```

**cascade**

```
\[ u \rightarrow S_1 \rightarrow w \rightarrow S_2 \rightarrow + \rightarrow y \]
```

An unknown input signal

\[ \eta \rightarrow + \rightarrow v \]

\[ \varepsilon \rightarrow + \rightarrow \]

\[ \varepsilon \rightarrow + \rightarrow \]

\[ \varepsilon \rightarrow + \rightarrow \]

The diagram illustrates systems with uncertain inputs, including errors-in-variables and Hammerstein systems. The cascade configuration is also shown, where an unknown input signal \( \eta \) is added to \( v \) before being fed into the system.
Systems with uncertain inputs

errors-in-variables

Hammerstein

cascade

blind
Systems with uncertain inputs

- A linear system $S$

- Errors-in-variables

- Hammerstein

- Cascade

- Blind
Systems with uncertain inputs

- A linear system $S$
- An unknown input signal $w$
The uncertain-input system

- Linear system $S$
- Unknown input $w$
The uncertain-input system

- Linear system $S$
- Unknown input $w$
- Prior knowledge
The uncertain-input system

- Linear system $S$
- Unknown input $w$
- Prior knowledge

How do we model these?
Gaussian processes
Gaussian processes

- Gaussian distribution over functions

\[ f(\cdot) \sim \mathcal{N}(\mu(\cdot), K(\cdot, \cdot)) \]
Gaussian processes

- Gaussian distribution over functions

\[ f(\cdot) \sim \mathcal{N}(\mu(\cdot), K(\cdot, \cdot)) \]

- The values of the function have a joint Gaussian distribution

\[
\begin{bmatrix}
    f(x_1) \\
    f(x_2) \\
    f(x_3)
\end{bmatrix} = \mathcal{N}
\begin{pmatrix}
    \mu(x_1) \\
    \mu(x_2) \\
    \mu(x_3)
\end{pmatrix},
\begin{pmatrix}
    K(x_1, x_1) & K(x_1, x_2) & K(x_1, x_3) \\
    K(x_2, x_1) & K(x_2, x_2) & K(x_2, x_3) \\
    K(x_3, x_1) & K(x_3, x_2) & K(x_3, x_3)
\end{pmatrix}
\]
Gaussian processes

- Gaussian distribution over functions

\[ f(\cdot) \sim \mathcal{N}(\mu(\cdot), K(\cdot, \cdot)) \]

- The values of the function have a joint Gaussian distribution

\[
\begin{bmatrix}
  f(x_1) \\
  f(x_2) \\
  f(x_3)
\end{bmatrix} = \mathcal{N}
\begin{pmatrix}
  \begin{bmatrix}
    \mu(x_1) \\
    \mu(x_2) \\
    \mu(x_3)
  \end{bmatrix},
  \begin{bmatrix}
    K(x_1, x_1) & K(x_1, x_2) & K(x_1, x_3) \\
    K(x_2, x_1) & K(x_2, x_2) & K(x_2, x_3) \\
    K(x_3, x_1) & K(x_3, x_2) & K(x_3, x_3)
  \end{bmatrix}
\end{pmatrix}
\]

- Given values of the function \( y = f(\tilde{x}) \), we can estimate

\[
\hat{f}(x) = \mathbb{E} \{ f(x) | y \} = \mu(x) + K(x, \tilde{x})[K(\tilde{x}, \tilde{x})]^{-1}(y - \mu(\tilde{x}))
\]
Gaussian processes
Gaussian processes
Gaussian processes
Gaussian processes
Gaussian processes
Gaussian processes
Gaussian processes
Gaussian processes
Using Gaussian processes to encode information

\[ K(x_1, x_2) = e^{-\frac{1}{\theta}(x_1-x_2)^2} \]
Using Gaussian processes to encode information

\[ K(x_1, x_2) = e^{-\frac{1}{\theta}(x_1-x_2)^2} \]

\[ \theta = 0.1 \]
Using Gaussian processes to encode information

\[ K(x_1, x_2) = e^{-\frac{1}{\theta}(x_1 - x_2)^2} \]

- \( \theta = 0.1 \)
- \( \theta = 0.01 \)
Using Gaussian processes to encode information

\[ K(x_1, x_2) = e^{-\frac{1}{\theta}(x_1 - x_2)^2} \]

- \( \theta = 0.1 \)
- \( \theta = 0.01 \)
- \( \theta = 1 \)
Modeling the uncertain input system

\[ g(\rho) \sim N(\mu_g(\rho), K_g(\rho)) \]

\[ \mu_g(\rho) = E\{g_i\}, \quad K_g(\rho)_{ij} = \text{cov}\{g_i, g_j\} \]

\[ w(\theta) \sim N(\mu_w(\theta), K_w(\theta)) \]

\[ \mu_w(\theta) = E\{w_i\}, \quad K_w(\theta)_{ij} = \text{cov}\{w_i, w_j\} \]
Modeling the uncertain input system

\[ g \sim \mathcal{N}(\mu_g(\rho), K_g(\rho)) \]

\[ [\mu_g(\rho)]_i = \mathbb{E}\{g_i\} \quad [K_g(\rho)]_{ij} = \text{cov}\{g_i, g_j\} \]
Modeling the uncertain input system

- Gaussian process prior for the impulse response
  \[ g \sim \mathcal{N}(\mu_g(\rho), K_g(\rho)) \]
  \[
  \begin{align*}
  \mu_g(\rho)_i &= E\{g_i\}, \\
  K_g(\rho)_{ij} &= \text{cov}\{g_i, g_j\}
  \end{align*}
  \]

- Gaussian process prior for the input
  \[ w \sim \mathcal{N}(\mu_w(\theta), K_w(\theta)) \]
  \[
  \begin{align*}
  \mu_w(\theta)_i &= E\{w_i\}, \\
  K_w(\theta)_{ij} &= \text{cov}\{w_i, w_j\}
  \end{align*}
  \]
The system is stable and LTI noiseless.

The noises are additive, Gaussian, and white:

\[ w = v + \eta \]
\[ y = g \ast w + \varepsilon \]
\[ \eta \sim N(0, \sigma^2_v) \]
\[ \varepsilon \sim N(0, \sigma^2_y) \]
Measurement setup

The system is stable and LTI

\[ y_{\text{noiseless}} = w \ast g \]
Measurement setup

- The system is stable and LTI
  \[ y_{\text{noiseless}} = w \ast g \]

- The noises are additive, Gaussian, and white
  \[ v = w + \eta \quad y = w \ast g + \varepsilon \]
  \[ \eta \sim \mathcal{N}(0, \sigma^2_v I) \quad \varepsilon \sim \mathcal{N}(0, \sigma^2_y I) \]
The uncertain-input model

\[
\begin{align*}
    y &= w \ast g + \varepsilon \\
    v &= w + \eta \\
    \varepsilon &\sim \mathcal{N}(0, \sigma_y^2 I) \\
    \eta &\sim \mathcal{N}(0, \sigma_v^2 I) \\
    g &\sim \mathcal{N}(\mu_g(\rho), K_g(\rho)) \\
    w &\sim \mathcal{N}(\mu_w(\theta), K_w(\theta))
\end{align*}
\]
Inference in uncertain-input models

\[ g \sim N(\mu_g(\rho), K_g(\rho)) \]

\[ w \sim N(\mu_w(\theta), K_w(\theta)) \]

We would like to have the conditional mean estimates

\[ \hat{g} = E\{g | v, y\} \]

\[ \hat{w} = E\{w | v, y\} \]

We do not know the hyperparameters

\[ \tau = \{\rho, \theta, \sigma^2_v, \sigma^2_y\} \]

\[ \hat{g} = \hat{g}(\tau) \]

\[ \hat{w} = \hat{w}(\tau) \]

We need to estimate them from data
Inference in uncertain-input models

- Bayesian assumption

\[ g \sim \mathcal{N}(\mu_g(\rho), K_g(\rho)) \quad w \sim \mathcal{N}(\mu_w(\theta), K_w(\theta)) \]

We would like to have the conditional mean estimates

\[ \hat{g} = E\{g | v, y\} \quad \hat{w} = E\{w | v, y\} \]

We do not know the hyperparameters

\[ \tau = \{\rho, \theta, \sigma^2_v, \sigma^2_y\} \]

\[ \hat{g} = \hat{g}(\tau) \quad \hat{w} = \hat{w}(\tau) \]

We need to estimate them from data
Inference in uncertain-input models

- Bayesian assumption

\[ g \sim \mathcal{N}(\mu_g(\rho), K_g(\rho)) \quad w \sim \mathcal{N}(\mu_w(\theta), K_w(\theta)) \]

- We would like to have the conditional mean estimates

\[ \hat{g} = \mathbb{E}\{g | v, y\} \quad \hat{w} = \mathbb{E}\{w | v, y\} \]
Inference in uncertain-input models

- Bayesian assumption

\[ g \sim \mathcal{N}(\mu_g(\rho), K_g(\rho)) \quad w \sim \mathcal{N}(\mu_w(\theta), K_w(\theta)) \]

- We would like to have the conditional mean estimates

\[ \hat{g} = \mathbb{E}\{g|\nu, y\} \quad \hat{w} = \mathbb{E}\{w|\nu, y\} \]

- We do not know the hyperparameters \( \tau = \{\rho, \theta, \sigma_v^2, \sigma_y^2\} \):

\[ \hat{g} = \hat{g}(\tau) \quad \hat{w} = \hat{w}(\tau) \]
Inference in uncertain-input models

- Bayesian assumption

\[ g \sim \mathcal{N}(\mu_g(\rho), K_g(\rho)) \quad w \sim \mathcal{N}(\mu_w(\theta), K_w(\theta)) \]

- We would like to have the conditional mean estimates

\[ \hat{g} = \mathbb{E}\{g|v, y\} \quad \hat{w} = \mathbb{E}\{w|v, y\} \]

- We do not know the hyperparameters \( \tau = \{\rho, \theta, \sigma_v^2, \sigma_y^2\} \)

\[ \hat{g} = \hat{g}(\tau) \quad \hat{w} = \hat{w}(\tau) \]

*We need to estimate them from data*
Empirical Bayes

We choose the hyperparameters that maximize the marginal likelihood

\[ \hat{\tau} = \arg \max_{\tau} p(y, v; \tau) \]
Empirical Bayes

- We choose the hyperparameters that maximize the *marginal likelihood*

\[
\hat{\tau} = \arg \max_\tau p(y, v; \tau)
\]
We choose the hyperparameters that maximize the marginal likelihood
\[ \hat{\tau} = \arg \max_{\tau} p(y, v; \tau) \]

Estimates:
\[ \hat{g}(\hat{\tau}) = \mathbb{E}\{g|v, y; \hat{\tau}\} \quad \hat{w}(\hat{\tau}) = \mathbb{E}\{w|v, y; \hat{\tau}\} \]
Is it really that simple?

Short answer

Yes

Long answer

Yes, but...

- we need to calculate expected values
- some distributions are not available in closed form
- we need to maximize the marginal likelihood
Is it really that simple?

Short answer

Yes
Is it really that simple?

Short answer

Yes

Long answer

Yes, but...
Is it really that simple?

<table>
<thead>
<tr>
<th>Short answer</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long answer</td>
<td>Yes, <em>but</em>…</td>
</tr>
<tr>
<td></td>
<td>- we need to calculate expected values</td>
</tr>
</tbody>
</table>
Is it really that simple?

Short answer

Yes

Long answer

Yes, but...

- we need to calculate expected values
- some distributions are not available in closed form
Is it really that simple?

<table>
<thead>
<tr>
<th>Short answer</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long answer</td>
<td><strong>Yes, but…</strong></td>
</tr>
<tr>
<td></td>
<td>- we need to calculate expected values</td>
</tr>
<tr>
<td></td>
<td>- some distributions are not available in closed form</td>
</tr>
<tr>
<td></td>
<td>- we need to maximize the marginal likelihood</td>
</tr>
<tr>
<td>Short answer</td>
<td>Yes</td>
</tr>
<tr>
<td>--------------</td>
<td>-----</td>
</tr>
</tbody>
</table>
| Long answer  | Yes, *but...*  
  - we need to calculate expected values ← Monte Carlo  
  - some distributions are not available in closed form  
  - we need to maximize the marginal likelihood |
Is it really that simple?

**Short answer**

Yes

**Long answer**

Yes, but...

- we need to calculate expected values ← Monte Carlo
- some distributions are not available in closed form ← Gibbs
- we need to maximize the marginal likelihood
Is it really that simple?

**Short answer**

Yes

**Long answer**

Yes, but...
- we need to calculate expected values ← Monte Carlo
- some distributions are not available in closed form ← Gibbs
- we need to maximize the marginal likelihood ← EM
Calculating the posterior mean

- We need the posterior means

\[ E \{ g | y, \nu; \tau \} \]

\[ E \{ w | y, \nu; \tau \} \]
Calculating the posterior mean

- We need the posterior means

\[ E\{g|y, \nu; \tau\} = \int g \ p(g, w|y, \nu; \tau) \ dw \ dg \]
\[ E\{w|y, \nu; \tau\} = \int w \ p(g, w|y, \nu; \tau) \ dg \ dw \]
Calculating the posterior mean

- We need the posterior means

\[ E \{ g \mid y, v; \tau \} = \int g \ p(g, w \mid y, v; \tau) \ dw \ dg \]

\[ E \{ w \mid y, v; \tau \} = \int w \ p(g, w \mid y, v; \tau) \ dg \ dw \]
Calculating the posterior mean

- We need the posterior means

\[
E \{g|y, v; \tau\} = \int g \ p(g, w|y, v; \tau) \ dw \ dg
\]

\[
E \{w|y, v; \tau\} = \int w \ p(g, w|y, v; \tau) \ dg \ dw
\]

*The joint distribution is problematic to compute!*
Calculating the posterior mean

- We need the posterior means

\[
E \{ g | y, v; \tau \} = \int g \, p(g, w | y, v; \tau) \, dw \, dg
\]

\[
E \{ w | y, v; \tau \} = \int w \, p(g, w | y, v; \tau) \, dg \, dw
\]

The joint distribution is problematic to compute!
Monte Carlo integration

We make a particle approximation of the distribution

\[ p(g, w | y, v; \tau) \approx \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \]
Monte Carlo integration

- We make a particle approximation of the distribution

\[
p(g, w|y, \nu; \tau) \approx \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j))
\]
Monte Carlo integration

- We make a particle approximation of the distribution

\[ p(g, w | y, v; \tau) \approx \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \]
Monte Carlo integration

- We make a particle approximation of the distribution

\[ p(g, w | y, v; \tau) \approx \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \]
Monte Carlo integration

- We make a particle approximation of the distribution

\[ p(g, w|y, v; \tau) \approx \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \]

- We approximate the posterior mean

\[ \mathbf{E} \{g|y, v\} = \int g \, p(g, w|y, v; \tau) \, dw \, dg \]
Monte Carlo integration

- We make a particle approximation of the distribution

\[ p(g, w|y, v; \tau) \approx \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \]

- We approximate the posterior mean

\[
\mathbb{E}\{g|y, v\} = \int g \ p(g, w|y, v; \tau) \ dw \ dg \\
\approx \int g \ \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \ dw \ dg
\]
Monte Carlo integration

- We make a particle approximation of the distribution

\[
p(g, w|y, v; \tau) \approx \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j))
\]

- We approximate the posterior mean

\[
E\{g|y, v\} = \int g \ p(g, w|y, v; \tau) \, dw \, dg
\]

\[
\approx \int g \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \, dw \, dg = \frac{1}{M} \sum_{j=1}^{M} \bar{g}(j)
\]
Monte Carlo integration

- We make a particle approximation of the distribution

\[ p(g, w \mid y, v; \tau) \approx \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \]

- We approximate the posterior mean

\[ \mathbb{E}\{g \mid y, v\} = \int g \; p(g, w \mid y, v; \tau) \; dw \; dg \]

\[ \approx \int g \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \; dw \; dg = \frac{1}{M} \sum_{j=1}^{M} \bar{g}(j) \]

- Similarly

\[ \mathbb{E}\{w \mid y, v\} = \int w \; p(g, w \mid y, v; \tau) \; dg \; dw \approx \frac{1}{M} \sum_{j=1}^{M} \bar{w}(j) \]
Particle approximation
Particle approximation

We want samples from a joint distribution $p(g,w|y,v;\tau)$ ← difficult to evaluate!

but $p(g|y,w;\tau)$ ← Gaussian!

$p(w|y,v,g;\tau)$ ← Gaussian!
We want samples from a joint distribution

\[ p(g, w | y, v; \tau) \]
Particle approximation

We want samples from a joint distribution

$$p(g, w | y, v; \tau) \leftarrow \text{difficult to evaluate!}$$
We want samples from a joint distribution

\[ p(g, w|y, v; \tau) \leftarrow \text{difficult to evaluate!} \]

but

\[ p(g|y, w; \tau) \]
Particle approximation

- We want samples from a joint distribution
  \[ p(g, w | y, v; \tau) \leftarrow \text{difficult to evaluate!} \]

- but

  \[ p(g | y, w; \tau) \leftarrow \text{Gaussian!} \]
We want samples from a joint distribution

\[ p(g, w | y, v; \tau) \leftarrow \text{difficult to evaluate!} \]

but

\[ p(g | y, w; \tau) \leftarrow \text{Gaussian!} \]
\[ p(w | y, v, g; \tau) \]
We want samples from a joint distribution

\[ p(g, w|y, v; \tau) \leftarrow \text{difficult to evaluate!} \]

but

\[ p(g|y, w; \tau) \leftarrow \text{Gaussian!} \]
\[ p(w|y, v, g; \tau) \leftarrow \text{Gaussian!} \]
The Gibbs sampler

g | y, w \sim \mathcal{N}

\( g \mid y, w \sim \mathcal{N} \)
The Gibbs sampler

\[
g | y, w \sim \mathcal{N}
\]

\[
\bar{g}^{(k+1)} \sim p(g | y, \bar{w}^{(k)}; \hat{\rho}, \hat{\sigma}_y^2),
\]
The Gibbs sampler

\[ g \mid y, w \sim \mathcal{N} \]

\[ w \mid g, v, y \sim \mathcal{N} \]

\[ \bar{g}^{(k+1)} \sim p(g \mid y, \bar{w}^{(k)}; \hat{\rho}, \hat{\sigma}_y^2), \]
The Gibbs sampler

\[ g | y, w \sim \mathcal{N} \]

\[ w | g, v, y \sim \mathcal{N} \]

\[ \bar{g}^{(k+1)} \sim p(g | y, \bar{w}^{(k)} ; \hat{\rho}, \hat{\sigma}^2_y), \]

\[ \bar{w}^{(k+1)} \sim p(w | y, v, \bar{g}^{(k+1)} ; \hat{\theta}, \hat{\sigma}^2) \]
The Gibbs sampler

\[ g | y, w \sim \mathcal{N} \]

\[ w | g, v, y \sim \mathcal{N} \]

\[ \bar{g}^{(k+1)} \sim p(g | y, \bar{w}^{(k)} ; \hat{\rho}, \hat{\sigma}_y^2), \]

\[ \bar{w}^{(k+1)} \sim p(w | y, v, \bar{g}^{(k+1)} ; \hat{\theta}, \hat{\sigma}^2) \]

\((\bar{w}^{(k)}, \bar{g}^{(k)})\) are samples from \(p(g, w | y, v ; \hat{\tau})\)
Gibbs sampling the UI model

\[
\begin{align*}
\{ (\bar{g}(0), \bar{w}(0)) \}
\end{align*}
\]
Gibbs sampling the UI model

\[
\left\{ (\bar{g}^{(0)}, \bar{w}^{(0)}), (\bar{g}^{(1)}, \bar{w}^{(1)}) \right\}
\]

\[
g | y, w \sim \mathcal{N}
\]
Gibbs sampling the UI model

\[
\begin{align*}
\{(\tilde{g}^{(0)}, \tilde{w}^{(0)}) & \quad (\tilde{g}^{(1)}, \tilde{w}^{(1)}) \}
\end{align*}
\]
Gibbs sampling the UI model

\[ \begin{align*}
\{ (\bar{g}^{(0)}, \bar{w}^{(0)}), &\quad (\bar{g}^{(1)}, \bar{w}^{(1)}), &\quad (\bar{g}^{(2)}, \bar{w}^{(2)}), \} \\
\end{align*} \]
Gibbs sampling the UI model

\[
\{ (\bar{g}^{(0)}, \bar{w}^{(0)}) \quad (\bar{g}^{(1)}, \bar{w}^{(1)}) \quad (\bar{g}^{(2)}, \bar{w}^{(2)}) \}
\]
Gibbs sampling the UI model

\[ g | y, w \sim \mathcal{N} \]

\[ \{ (\bar{g}^{(0)}, \bar{w}^{(0)}), (\bar{g}^{(1)}, \bar{w}^{(1)}), (\bar{g}^{(2)}, \bar{w}^{(2)}), (\bar{g}^{(3)}, \ldots) \} \]
Gibbs sampling the UI model

\[
\begin{align*}
\{ (\bar{g}^{(0)}, \bar{w}^{(0)}), (\bar{g}^{(1)}, \bar{w}^{(1)}), (\bar{g}^{(2)}, \bar{w}^{(2)}), (\bar{g}^{(3)}, \bar{w}^{(3)}) \}
\end{align*}
\]
Gibbs sampling the UI model

\[
\{(\bar{g}^{(0)}, \bar{w}^{(0)}), (\bar{g}^{(1)}, \bar{w}^{(1)}), (\bar{g}^{(2)}, \bar{w}^{(2)}), (\bar{g}^{(3)}, \bar{w}^{(3)}), (\bar{g}^{(4)}, \bar{w}^{(4)})\}
\]
Gibbs sampling the UI model

\[ ((\bar{g}^{(0)}, \bar{w}^{(0)}), (\bar{g}^{(1)}, \bar{w}^{(1)}), (\bar{g}^{(2)}, \bar{w}^{(2)}), (\bar{g}^{(3)}, \bar{w}^{(3)}), (\bar{g}^{(4)}, \bar{w}^{(4)})) \]
Maximizing the marginal likelihood

- We want to compute the marginal likelihood estimate

\[ \hat{\tau} = \arg \max_{\tau} p(y, \nu; \tau) \]
Maximizing the marginal likelihood

- We want to compute the *marginal likelihood estimate*

\[ \hat{\tau} = \arg \max_{\tau} p(y, \nu; \tau) \]

- but

\[ p(y, \nu; \tau) = \int p(y, \nu, g, w; \tau) \, dg \, dw \]
Maximizing the marginal likelihood

- We want to compute the *marginal likelihood estimate*

\[ \hat{\tau} = \arg \max_{\tau} p(y, v; \tau) \]

- but

\[
p(y, v; \tau) = \int p(y, v, g, w; \tau) \, dg \, dw \\
= \int p(y, v|g, w; \tau)p(g, w; \tau) \, dg \, dw
\]
Maximizing the marginal likelihood

- We want to compute the marginal likelihood estimate

$$\hat{\tau} = \arg \max_{\tau} p(y, v; \tau)$$

- but

$$p(y, v; \tau) = \int p(y, v, g, w; \tau) \, dg \, dw$$

$$= \int p(y, v|g, w; \tau)p(g, w; \tau) \, dg \, dw$$

A maximum likelihood problem with missing data!
EM for marginal likelihood estimation

- Maximum likelihood problem with missing data

\[ \hat{\tau} = \arg \max_{\tau} p(y, v; \tau) = \arg \max_{\tau} \int p(y, v, g, w; \tau) \, dg \, dw \]
EM for marginal likelihood estimation

- Maximum likelihood problem with missing data

\[
\hat{\tau} = \arg \max_{\tau} p(y, v; \tau) = \arg \max_{\tau} \int p(y, v, g, w; \tau) \, dg \, dw
\]

- We can use the EM-method!

**E-step** \( Q(\tau, \tau^{(k)}) = E \{ \log p(y, v, g, w; \tau) \} \)

w.r.t. \( p(g, w | y, v; \hat{\tau}^{(k)}) \)

**M-step** \( \tau^{(k+1)} = \arg \max Q(\tau, \tau^{(k)}) \)
**EM for marginal likelihood estimation**

- Maximum likelihood problem with missing data

\[ \hat{\tau} = \arg\max_{\tau} p(y, v; \tau) = \arg\max_{\tau} \int p(y, v, g, w; \tau) \, dg \, dw \]

- We can use the EM-method!

**E-step**

\[ Q(\tau, \tau^{(k)}) = E \{ \log p(y, v, g, w; \tau) \} \]

w.r.t. \( p(g, w|y, v; \hat{\tau}^{(k)}) \)

**M-step**

\[ \tau^{(k+1)} = \arg\max \, Q(\tau, \tau^{(k)}) \]
EM for marginal likelihood estimation

- Maximum likelihood problem with missing data

\[ \hat{\tau} = \arg \max_{\tau} p(y, v; \tau) = \arg \max_{\tau} \int p(y, v, g, w; \tau) \, dg \, dw \]

- We can use the EM-method!

**E-step**

\[ Q(\tau, \tau^{(k)}) = E \{ \log p(y, v, g, w; \tau) \} \]

w.r.t. \( p(g, w|y, v; \hat{\tau}^{(k)}) \)

**M-step**

\[ \tau^{(k+1)} = \arg \max Q(\tau, \tau^{(k)}) \]
EM for marginal likelihood estimation

- Maximum likelihood problem with missing data

\[ \hat{\tau} = \arg \max_{\tau} p(y, v; \tau) = \arg \max_{\tau} \int p(y, v, g, w; \tau) \, dg \, dw \]

- We can use the EM-method!

**E-step**

\[ Q(\tau, \tau^{(k)}) = E \{ \log p(y, v, g, w; \tau) \} \]

w.r.t. \( p(g, w | y, v; \hat{\tau}^{(k)}) \)

**M-step**

\[ \tau^{(k+1)} = \arg \max_{\tau} Q(\tau, \tau^{(k)}) \]
EM for marginal likelihood estimation

- Maximum likelihood problem with missing data

\[ \hat{\tau} = \arg \max_{\tau} p(y, v; \tau) = \arg \max_{\tau} \int p(y, v, g, w; \tau) \, dg \, dw \]

- We can use the EM-method!

**E-step**

\[ Q(\tau, \tau^{(k)}) = E \{ \log p(y, v, g, w; \tau) \} \]

w.r.t. \( p(g, w | y, v; \hat{\tau}^{(k)}) \)

**M-step**

\[ \tau^{(k+1)} = \arg \max Q(\tau, \tau^{(k)}) \]
EM for marginal likelihood estimation

- Maximum likelihood problem with missing data

\[ \hat{\tau} = \arg \max_{\tau} p(y, v; \tau) = \arg \max_{\tau} \int p(y, v, g, w; \tau) \, dg \, dw \]

- We can use the EM-method!

\[ Q(\tau, \tau^{(k)}) = E \{ \log p(y, v, g, w; \tau) \} \]

w.r.t. \( p(g, w | y, v; \hat{\tau}^{(k)}) \)

\[ \tau^{(k+1)} = \arg \max Q(\tau, \tau^{(k)}) \]
EM for marginal likelihood estimation

- Maximum likelihood problem with missing data

\[ \hat{\tau} = \arg\max_\tau p(y, v; \tau) = \arg\max_\tau \int p(y, v, g, w; \tau) \, dg \, dw \]

- We can use the EM-method!

  **E-step**
  \[ Q(\tau, \tau^{(k)}) = E \{ \log p(y, v, g, w; \tau) \} \]
  w.r.t. \( p(g, w|y, v; \hat{\tau}^{(k)}) \)

  **M-step**
  \[ \tau^{(k+1)} = \arg\max_{\tau} Q(\tau, \tau^{(k)}) \]
EM for marginal likelihood estimation

- Maximum likelihood problem with missing data

\[ \hat{\tau} = \arg \max_{\tau} p(y, v; \tau) = \arg \max_{\tau} \int p(y, v, g, w; \tau) \, dg \, dw \]

- We can use the EM-method!

**E-step**  
\[ Q(\tau, \tau^{(k)}) = E \{ \log p(y, v, g, w; \tau) \} \]

\[ \text{w.r.t. } p(g, w | y, v; \hat{\tau}^{(k)}) \]

**M-step**  
\[ \tau^{(k+1)} = \arg \max \, Q(\tau, \tau^{(k)}) \]
Monte Carlo Expectation Maximization

- The E-step is difficult to evaluate

\[ Q(\tau, \tau^{(k)}) = \mathbf{E}\left\{ \log p(y, v, g, w; \tau) \right\} \]
Monte Carlo Expectation Maximization

- The E-step is difficult to evaluate

\[ Q(\tau, \tau^{(k)}) = \mathbb{E}\{\log p(y, v, g, w; \tau)\} \]

\[ = \int \log [p(y, v, g, w; \tau)] p(g, w|y, v; \tau^{(k)}) dg \, dw \]
Monte Carlo Expectation Maximization

- The E-step is difficult to evaluate

\[ Q(\tau, \tau^{(k)}) = E \{ \log p(y, v, g, w; \tau) \} \]

\[ = \int \log [p(y, v, g, w; \tau)] p(g, w | y, v; \tau^{(k)}) \, dg \, dw \]
Monte Carlo Expectation Maximization

- The E-step is difficult to evaluate

\[
Q(\tau, \tau^{(k)}) = \mathbb{E}\{\log p(y, v, g, w; \tau)\}
= \int \log [p(y, v, g, w; \tau)] p(g, w|y, v; \tau^{(k)}) dg \, dw
\]

- We can use the same particle approximation as before!
Monte Carlo Expectation Maximization

- The E-step is difficult to evaluate

\[
Q(\tau, \tau^{(k)}) = \mathbf{E}\{\log p(y, v, g, w; \tau)\}
\]

\[
= \int \log [p(y, v, g, w; \tau)] p(g, w|y, v; \tau^{(k)}) dg \, dw
\]

- We can use the same particle approximation as before!

\[
Q(\tau, \tau^{(k)})
\]

\[
\approx \int \log [p(y, v, g, w; \tau)] \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) dg \, dw
\]
Monte Carlo Expectation Maximization

- The E-step is difficult to evaluate

\[
Q(\tau, \tau^{(k)}) = \mathbf{E}\{\log p(y, v, g, w; \tau)\}
= \int \log [p(y, v, g, w; \tau)] p(g, w|y, v; \tau^{(k)}) dg \, dw
\]

- We can use the same particle approximation as before!

\[
Q(\tau, \tau^{(k)}) 
\approx \int \log [p(y, v, g, w; \tau)] \frac{1}{M} \sum_{j=1}^{M} \delta(g - \bar{g}(j), w - \bar{w}(j)) \, dg \, dw
\]

\[
= \frac{1}{M} \sum_{j=1}^{M} \log [p(y, v, \bar{g}(j), \bar{w}(j); \tau)]
\]

Monte Carlo Expectation Maximization

MCEM for hyperparameter estimation
Monte Carlo Expectation Maximization

MCEM for hyperparameter estimation

MC-step  \[ \{g^{(j)}, w^{(j)}\}_{j=1}^M = \text{particle approximation of } p(g, w | y, v; \tau^{(k)}) \]
Monte Carlo Expectation Maximization

**MCEM for hyperparameter estimation**

**MC-step**\[ \{g^{(j)}, w^{(j)}\}_{j=1}^{M} = \text{particle approximation of } p(g, w|y, v; \tau^{(k)}) \]

**E-step**\[ \bar{Q}(\tau, \tau^{(k)}) = \frac{1}{M} \sum_{j=1}^{M} \log p(y, v, g^{(j)}, w^{(j)}; \tau) \]
Monte Carlo Expectation Maximization

MCEM for hyperparameter estimation

MC-step \( \{g^{(j)}, w^{(j)}\}_{j=1}^{M} = \) particle approximation of \( p(g, w | y, v; \tau^{(k)}) \)

E-step \( \bar{Q}(\tau, \tau^{(k)}) = \frac{1}{M} \sum_{j=1}^{M} \log p(y, v, g^{(j)}, w^{(j)}; \tau) \)

M-step \( \tau^{(k+1)} = \arg \max \bar{Q}(\tau, \tau^{(k)}) \)
Monte Carlo inference in UI models

MCEM for hyperparameter estimation

MC-step
\{ g(j), w(j) \}

\[ M_j = 1 \Rightarrow \text{GibbsSampler}(\tau(k)) \]

E-step
\[ \bar{Q}(\tau, \tau(k)) = 1 \]
\[ \frac{1}{M} \sum_{j=1}^{M} \log p(y, v, g(j), w(j); \tau) \]

M-step
\[ \tau(k+1) = \arg \max \bar{Q}(\tau, \tau(k)) \]

Posterior means
\{ \bar{g}(j), \bar{w}(j) \}

\[ \text{GibbsSampler}(\hat{\tau}) \]

\[ \mathbb{E}\{g|y, v\} \approx \frac{1}{M} \sum_{j=1}^{M} \bar{g}(j) \]

\[ \mathbb{E}\{w|y, v\} \approx \frac{1}{M} \sum_{j=1}^{M} \bar{w}(j) \]
Monte Carlo inference in UI models

- MCEM for hyperparameter estimation

MC-step \[ \{g^{(j)}, w^{(j)}\}_{j=1}^{M} = \text{GIBBS\_SAMPLER}(\tau^{(k)}) \]

E-step \[ \bar{Q}(\tau, \tau^{(k)}) = \frac{1}{M} \sum_{j=1}^{M} \log p(y, v, g^{(j)}, w^{(j)}; \tau) \]

M-step \[ \tau^{(k+1)} = \arg \max \bar{Q}(\tau, \tau^{(k)}) \]
Monte Carlo inference in UI models

- **MCEM for hyperparameter estimation**

**MC-step**
\[ \{g^{(j)}, w^{(j)}\}_{j=1}^M = \text{GIBBS}\text{SAMPLER}(\tau^{(k)}) \]

**E-step**
\[ \bar{Q}(\tau, \tau^{(k)}) = \frac{1}{M} \sum_{j=1}^{M} \log p(y, v, g^{(j)}, w^{(j)}; \tau) \]

**M-step**
\[ \tau^{(k+1)} = \arg\max \bar{Q}(\tau, \tau^{(k)}) \]

- **Posterior means**

\[ \{\bar{g}^{(j)}, \bar{w}^{(j)}\}_{j=1}^M = \text{GIBBS}\text{SAMPLER}(\hat{\tau}) \]

\[ E \{g|y, v\} \approx \frac{1}{M} \sum_{j=1}^{M} \bar{g}^{(j)} \]

\[ E \{w|y, v\} \approx \frac{1}{M} \sum_{j=1}^{M} \bar{w}^{(j)} \]
Example: Hammerstein models
Example: Hammerstein models

- Nonparametric Hammerstein model

\[ f(\cdot) \mid \mathcal{S} \]

\[ u \rightarrow f(\cdot) \rightarrow \mathcal{S} \rightarrow \varepsilon \rightarrow y \]
Example: Hammerstein models

- Nonparametric Hammerstein model

\[
u \xrightarrow{f(\cdot)} S \xrightarrow{\varepsilon} y\]
Example: Hammerstein models

- Nonparametric Hammerstein model

\[ f(\cdot) \]

- Uncertain-input model with

\[ g \sim \mathcal{N}(0, K_g(\rho)) \quad w \sim \mathcal{N}(0, K_w(\theta)) \]

\[ K_{g}(\rho)_{i,j} = \rho_{1}^{\max(i,j)} \]

\[ [K_{w}(\theta)]_{ij} = \exp \left[ -\frac{1}{\theta}(u_i - u_j)^2 \right] \]

\[ \text{Stable-spline kernel} \]
Example: Hammerstein models

\[ n_{lti} = 40, \quad N = 400, \quad SNR = 10, \quad \text{---} = \text{true, } - - - = \text{estimated} \]
Under the hood

- MCEM for hyperparameter estimation
MCEM for hyperparameter estimation

1. MC Step: Gibbs sampler
   - Reduce the covariance matrices with SVD (Bad conditioning!)
   - Draw $g$ and $w$
   - Discard 200 burn-in particles
   - Generate 500 particles for $w$ and $g$
Under the hood

- MCEM for hyperparameter estimation
  1. MC Step: Gibbs sampler
     ▶ Reduce the covariance matrices with SVD (Bad conditioning!)
     ▶ Draw $g$ and $w$
     ▶ Discard 200 burn-in particles
     ▶ Generate 500 particles for $w$ and $g$
  2. E-step and M-step available in (almost) closed form
Under the hood

- MCEM for hyperparameter estimation
  1. MC Step: Gibbs sampler
     - Reduce the covariance matrices with SVD (Bad conditioning!)
     - Draw $g$ and $w$
     - Discard 200 burn-in particles
     - Generate 500 particles for $w$ and $g$
  2. E-step and M-step available in (almost) closed form
  3. Iterate until convergence
Under the hood

- **MCEM for hyperparameter estimation**
  1. **MC Step: Gibbs sampler**
     - Reduce the covariance matrices with SVD (Bad conditioning!)
     - Draw $g$ and $w$
     - Discard 200 burn-in particles
     - Generate 500 particles for $w$ and $g$
  2. E-step and M-step available in (almost) closed form
  3. Iterate until convergence

- **Run new Gibbs sampler**
  1. Reduce the covariance matrices with SVD (Bad conditioning!)
  2. Discard 500 burn-in particles
  3. Generate 1000 particles for $w$ and $g$
Special classes of systems

\[ g \]

\[ w \rightarrow g \rightarrow y \]

\[ \eta \rightarrow v \]

\[ \epsilon \rightarrow y \]

PIPS: parametric-input parametric-system models

\[ K_g(\rho) = 0 \]

GIPS: Gaussian-input parametric-system models

\[ K_w(\theta) = 0 \]

PIGS: parametric-input Gaussian-system models

\[ EIGS: \text{Estimated-input Gaussian-system models} \]

\[ w = \hat{w} \]
Special classes of systems

- PIPS: parametric-input parametric-system models
  
  \[ K_g(\rho) = 0 \quad K_w(\theta) = 0 \]
Special classes of systems

- PIPS: parametric-input parametric-system models
  \[ K_g(\rho) = 0 \quad K_w(\theta) = 0 \]
- GIPS: Gaussian-input parametric-system models
  \[ K_g(\rho) = 0 \]
Special classes of systems

- PIPS: parametric-input parametric-system models
  \[ K_g(\rho) = 0 \quad K_w(\theta) = 0 \]
- GIPS: Gaussian-input parametric-system models
  \[ K_g(\rho) = 0 \]
- PIGS: parametric-input Gaussian-system models
  \[ K_w(\theta) = 0 \]
Special classes of systems

- **PIPS**: parametric-input parametric-system models
  \[ K_g(\rho) = 0 \quad K_w(\theta) = 0 \]
- **GIPS**: Gaussian-input parametric-system models
  \[ K_g(\rho) = 0 \]
- **PIGS**: parametric-input Gaussian-system models
  \[ K_w(\theta) = 0 \]
- **EIGS**: Estimated-input Gaussian-system models
  \[ w = \hat{w} \]
Special classes of systems

- **PIPS**: parametric-input parametric-system models
  \[ K_g(\rho) = 0 \quad K_w(\theta) = 0 \text{ ← ML and PEM!} \]

- **GIPS**: Gaussian-input parametric-system models
  \[ K_g(\rho) = 0 \]

- **PIGS**: parametric-input Gaussian-system models
  \[ K_w(\theta) = 0 \text{ ← Bayesian FIR models!} \]

- **EIGS**: Estimated-input Gaussian-system models
  \[ w = \hat{w} \]
Example: Semi-blind models
Example: Semi-blind models

- Semi-blind model

\[ u_t(\theta) \rightarrow S \rightarrow y \]

\[ \varepsilon \]

Piecewise constant input

\[ u_t(\theta) = T_1 T_2 T_3 T_4 \]

\[ \theta_1 \theta_2 \theta_3 \theta_4 \]

PIGS uncertain-input model with

\[ w \sim N(\theta, 0) \]

\[ g \sim N(0, K(\rho)) \]
Example: Semi-blind models

- Semi-blind model

\[ u_t(\theta) \rightarrow S \rightarrow y \]

- Piecewise constant input

\[ u_t(\theta) = \begin{cases} 
\theta_1 & \text{if } 1 < t < T_1 \\
\theta_2 & \text{if } T_1 < t < T_2 \\
\theta_3 & \text{if } T_2 < t < T_3 \\
\theta_4 & \text{if } T_3 < t < T_4 
\end{cases} \]

Equivalent to Bottegal, Risuleo and Hjalmarsson (2015)
Example: Semi-blind models

- Semi-blind model

- Piecewise constant input

\[ u_t(\theta) = \begin{cases} 
\theta_1 & \text{for } t < T_1 \\
\theta_2 & \text{for } T_1 \leq t < T_2 \\
\theta_3 & \text{for } T_2 \leq t < T_3 \\
\theta_4 & \text{for } T_3 \leq t < T_4 
\end{cases} \]

- PIGS uncertain-input model with

\[ w \sim \mathcal{N}(H\theta, 0) \quad \text{and} \quad g \sim \mathcal{N}(0, K(\rho)) \]
Example: Semi-blind models

\[ n_{\text{lti}} = 20, \; N = 200, \; p = 20, \; \text{SNR} = 10, \; \begin{cases} \text{---} & \text{true} \end{cases}, \; \begin{cases} \text{---} & \text{estimated} \end{cases} \]
Other examples

- Errors-in-variables
- Cascaded models
- Estimation of initial conditions
- Systems with missing data
Another Hammerstein example

\[ u \xrightarrow{f(\cdot)} w \xrightarrow{g} y \]

\[ \varepsilon \]

\[ \varepsilon \sim \mathcal{N}(0, \sigma^2) \]

\[ \varepsilon \sim 0.8 \mathcal{N}(0, \sigma^2) + 0.2 \mathcal{N}(0, \sigma^2) \]
Another Hammerstein example

\[ u \xrightarrow{f(\cdot)} w \xrightarrow{g} y + \varepsilon \]

\[ u \]

\[ f(u) \]

\[ g_k \]

\[ k \]

\[ \varepsilon \sim N(0, \sigma^2) \]

\[ \varepsilon \sim N(0.8, \sigma^2) + N(0.2, 10 \sigma^2) \]
Another Hammerstein example

\[ u \xrightarrow{f(\cdot)} w \xrightarrow{g} y \]

\[ \varepsilon \sim \mathcal{N}(0, \sigma^2) \]
Another Hammerstein example

\[ u \xrightarrow{f(\cdot)} w \xrightarrow{g} y \]

\[ \varepsilon \sim \mathcal{N}(0, \sigma^2) \]
\[ \varepsilon \sim 0.8\mathcal{N}(0, \sigma^2) + 0.2\mathcal{N}(0, 10\sigma^2) \]
Another Hammerstein example

\[ u \xrightarrow{f(\cdot)} w \xrightarrow{g} y \]

\[ \varepsilon \sim \mathcal{N}(0, \sigma^2) \quad \varepsilon \sim 0.8\mathcal{N}(0, \sigma^2) + 0.2\mathcal{N}(0, 10\sigma^2) \]
We need a noise model with *heavy tails*
Compounded Gaussian noise model

- Student-\(t\) model for the noise

\[ \varepsilon_t \sim \text{St}(\nu, \eta) \]
Compounded Gaussian noise model

- Student-\( t \) model for the noise

\[ \varepsilon_t \sim \text{St}(\nu, \eta) \]

- Equivalently, each noise precision is Gamma distributed

\[ \lambda_t \sim \text{Ga}(\alpha, \beta) \leftarrow \text{Prior density!} \]
Compounded Gaussian noise model

- Student-\( t \) model for the noise
  \[
  \varepsilon_t \sim \text{St}(\nu, \eta)
  \]

- Equivalently, each noise precision is Gamma distributed
  \[
  \lambda_t \sim \text{Ga}(\alpha, \beta) \quad \text{Prior density!}
  \]
  \[
  \varepsilon_t | \lambda_t \sim \mathcal{N}(0, \lambda_t^{-1})
  \]
Compounded Gaussian noise model

- Student-$t$ model for the noise
  \[ \varepsilon_t \sim \text{St}(\nu, \eta) \]

- Equivalently, each noise precision is Gamma distributed
  \[ \lambda_t \sim \text{Ga}(\alpha, \beta) \quad \leftarrow \quad \text{Prior density!} \]
  \[ \varepsilon_t | \lambda_t \sim \mathcal{N}(0, \lambda_t^{-1}) \]
Compounded Gaussian noise model

- Student-\(t\) model for the noise
  \[ \varepsilon_t \sim St(\nu, \eta) \]

- Equivalently, each noise precision is Gamma distributed
  \[ \lambda_t \sim Ga(\alpha, \beta) \quad \text{← Prior density!} \]
  \[ \varepsilon_t | \lambda_t \sim \mathcal{N}(0, \lambda_t^{-1}) \]
Model estimation

- Priors

\[ g \sim \mathcal{N}(0, K_g(\rho)), \quad w \sim \mathcal{N}(0, K_w(\rho)), \quad \lambda_t \sim \text{Ga}(\alpha, \beta) \]

\[ t = 1, \ldots, N \]

- Data model

\[ y | g, w, \lambda_1, \ldots, \lambda_N \sim \mathcal{N}(Wg, \text{Diag}\{\lambda_t^{-1}\}) \]
Model estimation

- Priors
  \[ g \sim \mathcal{N}(0, K_g(\rho)), \quad w \sim \mathcal{N}(0, K_w(\rho)), \quad \lambda_t \sim \text{Ga}(\alpha, \beta) \]
  \[ t = 1, \ldots, N \]

- Data model
  \[ y|g, w, \lambda_1, \ldots, \lambda_N \sim \mathcal{N}(Wg, \text{Diag}\{\lambda_t^{-1}\}) \]

- Same type of model as before only more hyperparameters
  \( \{\lambda_t\}_{t=1}^N \) instead of \( \sigma_y^2 \)
Model estimation

- **Priors**
  \[ g \sim \mathcal{N}(0, K_g(\rho)), \quad w \sim \mathcal{N}(0, K_w(\rho)), \quad \lambda_t \sim \text{Ga}(\alpha, \beta) \]
  \[ t = 1, \ldots, N \]

- **Data model**
  \[ y|g, w, \lambda_1, \ldots, \lambda_N \sim \mathcal{N}(Wg, \text{Diag}\{\lambda_t^{-1}\}) \]

- Same type of model as before only more hyperparameters
  \( \{\lambda_t\}_{t=1}^N \) instead of \( \sigma^2_y \)

- Same tools as before can be used (MCEM, Gibbs sampling)
Gibbs sampling

Result

A Markov chain with \((g, w, \lambda | y)\) as its stationary distribution

Known expressions for

\[ g | y, w, \lambda \sim \text{N}(m_g, P_g) \]

\[ w | y, g, \lambda \sim \text{N}(m_w, P_w) \]

\[ \lambda | y, g, w \sim \text{Ga}(\alpha_t, \beta_t) \]
Gibbs sampling

1. sample $\bar{g}(i) | \bar{w}(i-1), \bar{\lambda}(i-1), y$

2. sample $\bar{w}(i) | \bar{g}(i), \bar{\lambda}(i-1), y$

3. sample $\bar{\lambda}(i) | \bar{g}(i), \bar{w}(i), y$

Result: A Markov chain with $(g, w, \lambda | y)$ as its stationary distribution

Known expressions for:
- $g | y, w, \lambda \sim \mathcal{N}(\mu_g, \Sigma_g)$
- $w | y, g, \lambda \sim \mathcal{N}(\mu_w, \Sigma_w)$
- $\lambda | y, g, w \sim \text{Ga}(\alpha_t, \beta_t)$
Gibbs sampling

In sequence

1. sample $\bar{g}(i)$ | $\bar{w}(i-1)$, $\bar{\lambda}(i-1)$, $y$
2. sample $\bar{w}(i)$ | $\bar{g}(i)$, $\bar{\lambda}(i-1)$, $y$
3. sample $\bar{\lambda}(i)$ | $\bar{g}(i)$, $\bar{w}(i)$, $y$

Result

A Markov chain with $(g, w, \lambda | y)$ as its stationary distribution

Known expressions for

$g | y, w, \lambda \sim N(m_g, P_g)$

$w | y, g, \lambda \sim N(m_w, P_w)$

$\lambda | y, g, w \sim Ga(\alpha_t, \beta_t)$
Gibbs sampling

In sequence

1. sample $\tilde{g}^{(i)} | \tilde{w}^{(i-1)}, \tilde{\lambda}^{(i-1)}, y$
Gibbs sampling

In sequence
1. sample $\tilde{g}^{(i)}|\tilde{w}^{(i-1)}, \bar{\lambda}(i-1), y$
2. sample $\tilde{w}^{(i)}|\tilde{g}^{(i)}, \bar{\lambda}(i-1), y$
Gibbs sampling

In sequence
1. sample $\bar{g}^{(i)}|\bar{w}^{(i-1)}, \bar{\lambda}^{(i-1)}, y$
2. sample $\bar{w}^{(i)}|\bar{g}^{(i)}, \bar{\lambda}^{(i-1)}, y$
3. sample $\bar{\lambda}^{(i)}|\bar{g}^{(i)}, \bar{w}^{(i)}, y$

Result
A Markov chain with $(g, w, \lambda|y)$ as its stationary distribution

Known expressions for $g|y, w, \lambda \sim N(m_g, P_g)$
$w|y, g, \lambda \sim N(m_w, P_w)$
$\lambda|y, g, w \sim Ga(\alpha_t, \beta_t)$
Gibbs sampling

In sequence
1. sample $\tilde{g}^{(i)}|\tilde{w}^{(i-1)}, \tilde{\lambda}^{(i-1)}, y$
2. sample $\tilde{w}^{(i)}|\tilde{g}^{(i)}, \tilde{\lambda}^{(i-1)}, y$
3. sample $\tilde{\lambda}^{(i)}|\tilde{g}^{(i)}, \tilde{w}^{(i)}y$

Result
A Markov chain with $(g, w, \lambda|y)$ as its stationary distribution
Gibbs sampling

In sequence
1. sample $g^{(i)} | w^{(i-1)}, \lambda^{(i-1)}, y$
2. sample $w^{(i)} | g^{(i)}, \lambda^{(i-1)}, y$
3. sample $\lambda^{(i)} | g^{(i)}, w^{(i)}, y$

Result
A Markov chain with $(g, w, \lambda | y)$ as its stationary distribution

Known expressions for

$g | y, w, \lambda \sim \mathcal{N}(m_g, P_g)$
$w | y, g, \lambda \sim \mathcal{N}(m_w, P_w)$
$\lambda_t | y, g, w \sim \text{Ga}(\alpha_t, \beta_t)$
Approximate inference algorithm for Robust UI models

1: procedure Estimate-Robust(data)
2: Initialize $\hat{\rho}, \hat{\alpha}, \hat{\beta}$
3: while not converged do
4:     Approximate $Q(\rho, \alpha, \beta)$ \hspace{1cm} \triangleright \text{EM}
5:     $\hat{\rho}, \hat{\alpha}, \hat{\beta} \leftarrow \arg \max_{\rho, \alpha, \beta} Q(\rho, \alpha, \beta)$ \hspace{1cm} \triangleright \text{Gibbs sampling}
6: end while
7: $\hat{g}, \hat{w}, \hat{\lambda} \leftarrow \mathbb{E} \{ [g, w, \lambda | y; \hat{\rho}, \hat{\alpha}, \hat{\beta}] \}$ \hspace{1cm} \triangleright \text{Gibbs sampling}
8: return $\hat{g}, \hat{w}, \hat{\lambda}$
9: end procedure
Simulation study: Hammerstein systems

\[ u \xrightarrow{f(\cdot)} w \xrightarrow{g} y + \varepsilon \]

Polynomial nonlinearity of order \( p \in \{5, \ldots, 10\} \)

Transfer function of order \( m \in \{3, 4, 5\} \)

\( N = 300 \) samples of output, uniform white input in \([-1, 1]\)

Two methods:

- H-Gaussian with Gaussian noise model
- H-Robust with Student-\( t \) noise model

Two experiments:

- Varying number of outliers (with fixed variance 10\( \sigma^2 \))
- Varying outlier variance (with fixed fraction 15\%)
Simulation study: Hammerstein systems

\[ u \xrightarrow{f(\cdot)} w \xrightarrow{g} y + \varepsilon \]

- Polynomial nonlinearity of order \( p \in 5, \ldots, 10 \)
Simulation study: Hammerstein systems

- Polynomial nonlinearity of order $p \in 5, \ldots, 10$
- Transfer function of order $m \in 3, 4, 5$
Simulation study: Hammerstein systems

- Polynomial nonlinearity of order $p \in 5, \ldots, 10$
- Transfer function of order $m \in 3, 4, 5$
- $N = 300$ samples of output, uniform white input in $-1, 1$

![Diagram of Hammerstein system]

\[ u \xrightarrow{f(\cdot)} w \xrightarrow{g} y + \varepsilon \]
Simulation study: Hammerstein systems

- Polynomial nonlinearity of order $p \in 5, \ldots, 10$
- Transfer function of order $m \in 3, 4, 5$
- $N = 300$ samples of output, uniform white input in $-1, 1$

Two methods

Two experiments
- Varying number of outliers (with fixed variance $10\sigma^2$)
- Varying outlier variance (with fixed fraction 15%)
Simulation study: Hammerstein systems

- Polynomial nonlinearity of order $p \in 5, \ldots, 10$
- Transfer function of order $m \in 3, 4, 5$
- $N = 300$ samples of output, uniform white input in $-1, 1$

Two methods

**H-Gaussian** with Gaussian noise model

**H-Robust** with Student-$t$ noise model

Two experiments

- Varying number of outliers (with fixed variance $10 \sigma^2$)
- Varying outlier variance (with fixed fraction 15%)
Simulation study: Hammerstein systems

- Polynomial nonlinearity of order $p \in 5, \ldots, 10$
- Transfer function of order $m \in 3, 4, 5$
- $N = 300$ samples of output, uniform white input in $-1, 1$

Two methods

- **H-Gaussian** with Gaussian noise model
- **H-Robust** with Student-$t$ noise model
Simulation study: Hammerstein systems

- Polynomial nonlinearity of order $p \in 5, \ldots, 10$
- Transfer function of order $m \in 3, 4, 5$
- $N = 300$ samples of output, uniform white input in $-1, 1$

Two methods
- **H-Gaussian** with Gaussian noise model
- **H-Robust** with Student-$t$ noise model

Two experiments

\[ u \xrightarrow{f(\cdot)} w \xrightarrow{g} y \]
Simulation study: Hammerstein systems

Polynomial nonlinearity of order \( p \in 5, \ldots, 10 \)
Transfer function of order \( m \in 3, 4, 5 \)
\( N = 300 \) samples of output, uniform white input in \(-1, 1\)

Two methods

H-Gaussian with Gaussian noise model
H-Robust with Student-\( t \) noise model

Two experiments

Varying number of outliers (with fixed variance \( 10\sigma^2 \))
Simulation study: Hammerstein systems

• Polynomial nonlinearity of order $p \in 5, \ldots, 10$
• Transfer function of order $m \in 3, 4, 5$
• $N = 300$ samples of output, uniform white input in $-1, 1$

Two methods

H-Gaussian with Gaussian noise model

H-Robust with Student-\textit{t} noise model

Two experiments

• Varying number of outliers (with fixed variance $10\sigma^2$)
• Varying outlier variance (with fixed fraction 15%)
Outlier fraction results

![Graph showing the relationship between the fraction of outliers and fit for different methods. The graph includes a line for 'g', a dashed line for 'f', and symbols for 'H-Robust' and 'H-Gaussian'.]
Outlier variance results

![Graph showing outlier variance results with fit curves for g and f, and markers for H-Robust and H-Gaussian.]
Conclusions
Conclusions

- Many classical problems are uncertain-inputs problems
Conclusions

- Many classical problems are uncertain-inputs problems
- Proposed a model for uncertain-input systems
Conclusions

- Many classical problems are uncertain-inputs problems
- Proposed a model for uncertain-input systems
- Estimated the parameters
  - Gibbs sampling
  - MCEM
  - Monte Carlo integration
Conclusions

- Many classical problems are uncertain-inputs problems
- Proposed a model for uncertain-input systems
- Estimated the parameters
  - Gibbs sampling
  - MCEM
  - Monte Carlo integration
- Examples of classical problems
  - Hammerstein model
  - Semi-blind model
Conclusions

- Many classical problems are uncertain-inputs problems
- Proposed a model for uncertain-input systems
- Estimated the parameters
  - Gibbs sampling
  - MCEM
  - Monte Carlo integration
- Examples of classical problems
  - Hammerstein model
  - Semi-blind model
- Proposed a robust extension with Student-\(t\) likelihood