Paremeter Estimation in Dynamical Systems

Scientific Computing for Systematic Model Building

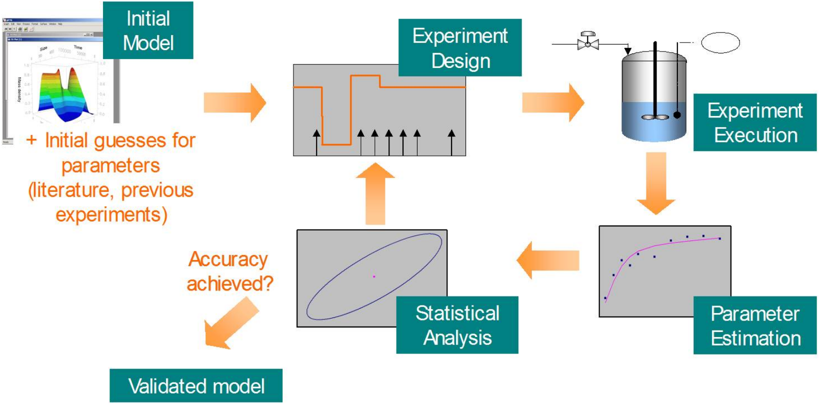
John Bagterp Jørgensen

Department of Applied Mathematics and Computer Science
Technical University of Denmark

02610 Optimization and Data Fitting

Mathematical Model Building

The Model Building Cycle



Deterministic Continuous-Discrete Dynamical Model

- Ordinary Differential Equations (ODEs) and output equation

$$\begin{aligned} x(t_0) &= \hat{x}_0 \\ \frac{dx}{dt}(t) &= f(x(t), u(t), d(t), \theta) \\ y(t_k) &= g(x(t_k), \theta) \end{aligned}$$

- Reformulation

$$\begin{aligned} x(t_0) &= \hat{x}_0 \\ dx(t) &= f(x(t), u(t), d(t), \theta)dt \\ y(t_k) &= g(x(t_k), \theta) \end{aligned}$$

- Explicit Euler Discretization

$$\begin{aligned} x_0 &= \hat{x}_0 \\ x_{k+1} &= x_k + f(x_k, u_k, d_k, \theta)\Delta t = F(x_k, u_k, d_k, \theta) \\ y_k &= g(x_k, \theta) \end{aligned}$$

Stochastic Continuous-Discrete Dynamical Model

- Ordinary Differential Equations (ODEs) and output equation

$$
\begin{aligned}
x(t_0) &= \hat{x}_0 \\
dx(t) &= f(x(t), u(t), d(t), \theta)dt \\
y(t_k) &= g(x(t_k), \theta)
\end{aligned}
$$

- Stochastic Differential Equations (SDEs) and output equation

$$
\begin{aligned}
\boldsymbol{x}(t_0) &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
d\boldsymbol{x}(t) &= \overbrace{f(\boldsymbol{x}(t), u(t), d(t), \theta)dt}^{=\text{drift}} + \overbrace{\sigma(\boldsymbol{x}(t), u(t), d(t), \theta)d\boldsymbol{\omega}(t)}^{=\text{diffusion}} & d\boldsymbol{\omega}(t) &\sim N_{iid}(0, Idt) \\
\boldsymbol{y}(t_k) &= g(\boldsymbol{x}(t_k), \theta) + \boldsymbol{v}(t_k) & \boldsymbol{v}(t_k) &\sim N_{iid}(0, R(\theta))
\end{aligned}
$$

- Euler-Maruyama Discretization (Explicit-Explicit)

$$
\begin{aligned}
\boldsymbol{x}_0 &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + f(\boldsymbol{x}_k, u_k, d_k, \theta)\Delta t + \sigma(\boldsymbol{x}_k, u_k, d_k, \theta)\Delta\boldsymbol{\omega}_k & \Delta\boldsymbol{\omega}_k &\sim N_{iid}(0, I\Delta t) \\
\boldsymbol{y}_k &= g(\boldsymbol{x}_k, \theta) + \boldsymbol{v}_k & \boldsymbol{v}_k &\sim N_{iid}(0, R(\theta))
\end{aligned}
$$

- Stochastic Differential Equations (SDEs) and output equation

$$
\begin{aligned}
\boldsymbol{x}(t_0) &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
d\boldsymbol{x}(t) &= f(\boldsymbol{x}(t), u(t), d(t), \theta)dt + \sigma(\boldsymbol{x}(t), u(t), d(t), \theta)d\boldsymbol{\omega}(t) & d\boldsymbol{\omega}(t) &\sim N_{iid}(0, Idt) \\
\boldsymbol{y}(t_k) &= g(\boldsymbol{x}(t_k), \theta) + \boldsymbol{v}(t_k) & \boldsymbol{v}(t_k) &\sim N_{iid}(0, R(\theta))
\end{aligned}
$$

- Euler-Maruyama Discretization (Explicit-Explicit)

$$
\begin{aligned}
\boldsymbol{x}_0 &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + f(\boldsymbol{x}_k, u_k, d_k, \theta)\Delta t + \sigma(\boldsymbol{x}_k, u_k, d_k, \theta)\Delta\boldsymbol{\omega}_k & \Delta\boldsymbol{\omega}_k &\sim N_{iid}(0, I\Delta t) \\
\boldsymbol{y}_k &= g(\boldsymbol{x}_k, \theta) + \boldsymbol{v}_k & \boldsymbol{v}_k &\sim N_{iid}(0, R(\theta))
\end{aligned}
$$

- Discretized system

$$
\begin{aligned}
\boldsymbol{x}_0 &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
\boldsymbol{x}_{k+1} &= F(\boldsymbol{x}_k, u_k, d_k, \boldsymbol{w}_k, \theta) & \boldsymbol{w}_k &\sim N_{iid}(0, Q) \\
\boldsymbol{y}_k &= g(\boldsymbol{x}_k, \theta) + \boldsymbol{v}_k & \boldsymbol{v}_k &\sim N_{iid}(0, R(\theta))
\end{aligned}
$$

with

$$
\begin{aligned}
F(\boldsymbol{x}_k, u_k, d_k, \boldsymbol{w}_k, \theta) &= \boldsymbol{x}_k + f(\boldsymbol{x}_k, u_k, d_k, \theta)\Delta t + \sigma(\boldsymbol{x}_k, u_k, d_k, \theta)\boldsymbol{w}_k \\
\boldsymbol{w}_k &= \Delta\boldsymbol{\omega}_k \sim N_{iid}(0, I\Delta t) = N_{iid}(0, Q), \; Q = I\Delta t
\end{aligned}
$$

- Stochastic Differential Equations (SDEs) and output equation

$$
\begin{aligned}
\boldsymbol{x}(t_0) &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
d\boldsymbol{x}(t) &= f(\boldsymbol{x}(t), u(t), d(t), \theta)dt + \sigma(\boldsymbol{x}(t), u(t), d(t), \theta)d\boldsymbol{\omega}(t) & d\boldsymbol{\omega}(t) &\sim N_{iid}(0, Idt) \\
\boldsymbol{y}(t_k) &= g(\boldsymbol{x}(t_k), \theta) + \boldsymbol{v}(t_k) & \boldsymbol{v}(t_k) &\sim N_{iid}(0, R(\theta))
\end{aligned}
$$

- Euler-Maruyama Discretization (Explicit-Explicit)

$$
\begin{aligned}
\boldsymbol{x}_0 &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + f(\boldsymbol{x}_k, u_k, d_k, \theta)\Delta t + \sigma(\boldsymbol{x}_k, u_k, d_k, \theta)\Delta\boldsymbol{\omega}_k & \Delta\boldsymbol{\omega}_k &\sim N_{iid}(0, I\Delta t) \\
\boldsymbol{y}_k &= g(\boldsymbol{x}_k, \theta) + \boldsymbol{v}_k & \boldsymbol{v}_k &\sim N_{iid}(0, R(\theta))
\end{aligned}
$$

- Discretized system

$$
\begin{aligned}
\boldsymbol{x}_0 &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
\boldsymbol{x}_{k+1} &= F(\boldsymbol{x}_k, u_k, d_k, \theta) + \boldsymbol{w}_k, & \boldsymbol{w}_k &\sim N_{iid}(0, Q_k(\theta)) \\
\boldsymbol{y}_k &= g(\boldsymbol{x}_k, \theta) + \boldsymbol{v}_k & \boldsymbol{v}_k &\sim N_{iid}(0, R(\theta))
\end{aligned}
$$

with

$$
\begin{aligned}
F(\boldsymbol{x}_k, u_k, d_k, \boldsymbol{w}_k, \theta) &= \boldsymbol{x}_k + f(\boldsymbol{x}_k, u_k, d_k, \theta)\Delta t \\
\boldsymbol{w}_k &= [\sigma(\boldsymbol{x}_k, u_k, d_k, \theta)\Delta\boldsymbol{\omega}_k] \sim N_{iid}(0, Q_k(\theta)) \\
Q_k(\theta) &= \sigma(\boldsymbol{x}_k, u_k, d_k, \theta)\,[I\Delta t]\,\sigma(\boldsymbol{x}_k, u_k, d_k, \theta)' \\
&= \left[\sigma(\boldsymbol{x}_k, u_k, d_k, \theta)\sigma(\boldsymbol{x}_k, u_k, d_k, \theta)'\right]\Delta t
\end{aligned}
$$

Filtering and Prediction

Extended Kalman Filter (EKF)

- Discrete-time model

$$
\begin{aligned}
\boldsymbol{x}_0 &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
\boldsymbol{x}_{k+1} &= F(\boldsymbol{x}_k, u_k, d_k, \theta) + \boldsymbol{w}_k, & \boldsymbol{w}_k &\sim N_{iid}(0, Q_k) \quad Q_k = Q_k(\theta) \\
\boldsymbol{y}_k &= g(\boldsymbol{x}_k, \theta) + \boldsymbol{v}_k & \boldsymbol{v}_k &\sim N_{iid}(0, R_k) \quad R_k = R(\theta)
\end{aligned}
$$

- Extended Kalman Filter Algorithm ($\hat{x}_{0|-1} = \hat{x}_0$, $P_{0|-1} = \hat{P}_0$)
  - Measurement update

$$
\begin{aligned}
\hat{y}_{k|k-1} &= g(\hat{x}_{k|k-1}, \theta) & C_k &= \frac{\partial g}{\partial x}(\hat{x}_{k|k-1}, \theta) \\
e_k &= y_k - \hat{y}_{k|k-1} & R_{e,k} &= C_k P_{k|k-1} C_k' + R_k \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k e_k & K_k &= P_{k|k-1} C_k' R_{e,k}^{-1}
\end{aligned}
$$

$$
P_{k|k} = P_{k|k-1} - K_k R_{e,k} K_k' = (I - K_k C_k) P_{k|k-1} (I - K_k C_k)' + K_k R_k K_k'
$$

  - Time update (One-step prediction)

$$
\begin{aligned}
\hat{x}_{k+1|k} &= F(\hat{x}_{k|k}, u_k, d_k, \theta) \\
P_{k+1|k} &= A_k P_{k|k} A_k' + Q_k \qquad A_k = \frac{\partial F}{\partial x}(\hat{x}_{k|k}, u_k, d_k, \theta)
\end{aligned}
$$

Continuous-Discrete Extended Kalman Filter (CDEKF)

- Continuous-Discrete Stochastic Model

$$\begin{aligned}
\boldsymbol{x}(t_0) &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
d\boldsymbol{x}(t) &= f(\boldsymbol{x}(t), u(t), d(t), \theta)dt + \sigma(\boldsymbol{x}(t), u(t), d(t), \theta)d\boldsymbol{\omega}(t) & d\boldsymbol{\omega}(t) &\sim N_{iid}(0, Idt) \\
\boldsymbol{y}(t_k) &= g(\boldsymbol{x}(t_k), \theta) + \boldsymbol{v}(t_k) & \boldsymbol{v}(t_k) &\sim N_{iid}(0, R(\theta))
\end{aligned}$$

- Continuous-Discrete Extended Kalman Filter Algorithm ($\hat{x}_{0|-1} = \hat{x}_0$, $P_{0|-1} = \hat{P}_0$)
  - Measurement update

$$\begin{aligned}
\hat{y}_{k|k-1} &= g(\hat{x}_{k|k-1}, \theta) & C_k &= \frac{\partial g}{\partial x}(\hat{x}_{k|k-1}, \theta) \\
e_k &= y_k - \hat{y}_{k|k-1} & R_{e,k} &= C_k P_{k|k-1} C_k' + R_k \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k e_k & K_k &= P_{k|k-1} C_k' R_{e,k}^{-1} \\
P_{k|k} &= P_{k|k-1} - K_k R_{e,k} K_k' = (I - K_k C_k) P_{k|k-1} (I - K_k C_k)' + K_k R_k K_k'
\end{aligned}$$

  - Time update - compute $\hat{x}_{k+1|k} = \hat{x}_k(t_{k+1})$ and $P_{k+1|k} = P_k(t_{k+1})$ by solving

$$\begin{aligned}
\frac{d}{dt}\hat{x}_k(t) &= f(\hat{x}_k(t), u_k, d_k, \theta) & \hat{x}_k(t_k) &= \hat{x}_{k|k} \\
\frac{d}{dt}P_k(t) &= A_k(t)P_k(t) + P_k(t)A_k(t)' + \sigma_k(t)\sigma_k(t)' & P_k(t_k) &= P_{k|k} \\
A_k(t) &= \frac{\partial f}{\partial x}(\hat{x}_k(t), u_k, d_k, \theta) \\
\sigma_k(t) &= \sigma(\hat{x}_k(t), u_k, d_k, \theta)
\end{aligned}$$

Filters and Predictors

- Discrete Stochastic Model

$$
\begin{aligned}
\boldsymbol{x}_0 &= \hat{\boldsymbol{x}}_0 & \hat{\boldsymbol{x}}_0 &\sim N(\hat{x}_0, \hat{P}_0) \\
\boldsymbol{x}_{k+1} &= F(\boldsymbol{x}_k, u_k, d_k, \theta) + \boldsymbol{w}_k, & \boldsymbol{w}_k &\sim N_{iid}(0, Q_k) \quad Q_k = Q_k(\theta) \\
\boldsymbol{y}_k &= g(\boldsymbol{x}_k, \theta) + \boldsymbol{v}_k & \boldsymbol{v}_k &\sim N_{iid}(0, R_k) \quad R_k = R(\theta)
\end{aligned}
$$

  - Extended Kalman Filter (EKF)
  - Unscented Kalman Filter (UKF)
  - Ensemble Kalman Filter (EnKF)
  - Particle Filter (PF)
- Continuous-Discrete Stochastic Model

$$
\begin{aligned}
&\boldsymbol{x}(t_0) = \hat{\boldsymbol{x}}_0 & &\hat{\boldsymbol{x}}_0 \sim N(\hat{x}_0, \hat{P}_0) \\
&d\boldsymbol{x}(t) = f(\boldsymbol{x}(t), u(t), d(t), \theta)dt + \sigma(\boldsymbol{x}(t), u(t), d(t), \theta)d\boldsymbol{\omega}(t) & &d\boldsymbol{\omega}(t) \sim N_{iid}(0, Idt) \\
&\boldsymbol{y}(t_k) = g(\boldsymbol{x}(t_k), \theta) + \boldsymbol{v}(t_k) & &\boldsymbol{v}(t_k) \sim N_{iid}(0, R(\theta))
\end{aligned}
$$

  - Continuous-Discrete Extended Kalman Filter (CDEKF)
  - Continuous-Discrete Unscented Kalman Filter (CDUKF)
  - Continuous-Discrete Ensemble Kalman Filter (CDEnKF)
  - Continuous-Discrete Particle Filter (CDPF)

Innovation

In the measurement update of the filters,
we compute the innovation and its covariance

$$e_k = e_k(\theta)$$
$$R_{e,k} = R_{e,k}(\theta)$$

The innovation is assumed to be distributed as

$$\boldsymbol{e}_k \sim N_{iid}(0, R_{e,k})$$

Statistical analysis is based on statistical tests assuming that the innovation has this distribution

Maximum-Likelihood Estimation

- Actual measurements $\{y_0, y_1, \ldots, y_{N_d}\}$
- Normally distributed independent variables

$$\boldsymbol{y}_k \sim N_{iid}(\hat{y}_k(\theta), R_k(\theta))$$

- Multivariate normal distrubtion

$$p_{y_k}(y_k; \theta) = \frac{1}{(2\pi)^{n_y/2} \left[\det R_k(\theta)\right]^{1/2}} \exp\left(-\frac{1}{2}(y_k - \hat{y}_k(\theta)) \left[R_k(\theta)\right]^{-1} (y_k - \hat{y}_k(\theta))\right)$$

$$p(\{y_k\}_{k=0}^{N_d}; \theta) = \prod_{k=0}^{N_d} p_{y_k}(y_k; \theta)$$

- Maximum Likelihood (ML) Estimation

$$\max_{\theta} \quad p(\{y_k\}_{k=0}^{N_d}; \theta) = \prod_{k=0}^{N_d} p_{y_k}(y_k; \theta)$$

- Negative log-likelihood estimation (equiv to maximum likelihood estimation)

$$L_k(\theta) = -\ln p_{y_k}(y_k; \theta) = \frac{n_y}{2} \ln(2\pi) + \frac{1}{2} \ln\left[\det R_k(\theta)\right] + \frac{1}{2}(y_k - \hat{y}_k(\theta)) \left[R_k(\theta)\right]^{-1} (y_k - \hat{y}_k(\theta))$$

$$\begin{aligned} L(\theta) &= -\ln p(\{y_k\}_{k=0}^{N_d}; \theta) = \sum_{k=0}^{N_d} L_k(\theta) \\ &= \frac{1}{2}\left(\sum_{k=0}^{N_d} \ln\left[\det R_k(\theta)\right] + \frac{1}{2}(y_k - \hat{y}_k(\theta)) \left[R_k(\theta)\right]^{-1} (y_k - \hat{y}_k(\theta))\right) + \frac{(N_d+1)n_y}{2} \ln(2\pi) \end{aligned}$$

$$\min_{\theta} \; L(\theta)$$

System Identification Methods

- Prediction-Error-Method (PEM)
  - Assume a stochastic model (discrete or continuous-discrete)
  - Compute the innovation and its covariance
    by a filter and prediction algorithm

$$
\begin{aligned}
e_k &= e_k(\theta) \\
R_{e,k} &= R_{e,k}(\theta)
\end{aligned}
$$

  - Assume that $\boldsymbol{e}_k \sim N_{iid}(0, R_{e,k})$ such that

$$
\begin{aligned}
V_{ML}(\theta) = \frac{1}{2} \sum_{k=0}^{N_d} \ln(\det R_{e,k}(\theta)) + e_k(\theta)' \left[R_{e,k}(\theta)\right]^{-1} e_k(\theta) \\
+ \frac{(N_d + 1) n_y}{2} \ln(2\pi)
\end{aligned}
$$

- Output-Error (OE)
  - Assume a deterministic model, but with measurement noise.
  - This is equivalent to a stochastic model with no process noise (diffusion) and perfectly known initial conditions. A PEM can be applied to such a system.
  - This is also know as a **simulation** model.

Parameter Estimation

$$\begin{aligned} \min_{\theta} \quad & V(\theta) \\ s.t. \quad & \theta_{\min} \leq \theta \leq \theta_{\max} \end{aligned}$$

Innovation (computed from model and data using a filter and predictor)

$$\begin{aligned} e_k(\theta) &= e_k \\ R_{e,k}(\theta) &= R_{e,k} \end{aligned}$$

Least squares (LS) objective function

$$V_{LS}(\theta) = \frac{1}{2} \sum_{k=0}^{N_d} \| e_k(\theta) \|_2^2$$

Maximum likelihood (ML) objective function

$$\begin{aligned} V_{ML}(\theta) = \frac{1}{2} \sum_{k=0}^{N_d} & \ln(\det R_{e,k}(\theta)) + e_k(\theta)' \left[R_{e,k}(\theta) \right]^{-1} e_k(\theta) \\ & + \frac{(N_d + 1) n_y}{2} \ln(2\pi) \end{aligned}$$

Maximum a posteriori (MAP) objective function

$$V_{MAP}(\theta) = V_{ML}(\theta) + \frac{1}{2} (\theta - \theta_0)' P_{\theta_0}^{-1} (\theta - \theta_0) + \frac{1}{2} \ln(\det P_{\theta_0}) + \frac{n_\theta}{2} \ln(2\pi)$$

Parameter Estimation - Bound Constrained Optimization

$$\begin{aligned} \min_{\theta} \quad & V(\theta) \\ s.t. \quad & \theta_{\min} \leq \theta \leq \theta_{\max} \end{aligned}$$

is solved by

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ s.t. \quad & l \leq x \leq u \end{aligned}$$

```
FMINCON finds a constrained minimum of a function of several variables.
FMINCON attempts to solve problems of the form:
min F(X)  subject to:  A*X  <= B, Aeq*X  = Beq (linear constraints)
X                     C(X) <= 0, Ceq(X) = 0   (nonlinear constraints)
LB <= X <= UB        (bounds)

xopt = fmincon(@fun, x0, [], [], [], [], lb, ub)
```

Parameter Estimation

Parameter Estimation

$$\min_x \; f(x)$$

- Model / prediction: $\hat{y}(x)$
- Measurement: y
- Error (residual): $e = e(x) = y - \hat{y}(x)$
- Covariance of error (residual): $R = R(x)$
- Objective function: $f(x)$
  - Least Squares (LS)

$$f(x) = \frac{1}{2} \left\| e(x) \right\|_2^2$$

  - Maximum Likelihood (ML) [negative log likelihood function]

$$f(x) = \frac{1}{2} \ln \left[\det R(x) \right] + \frac{1}{2} e(x)' R(x)^{-1} e(x)$$

- Error (residual): $e(x)$

$$e(x) = \begin{bmatrix} e_1(x) \\ \vdots \\ e_m(x) \end{bmatrix}$$

$$J(x) = \frac{\partial e}{\partial x}(x) = \begin{bmatrix} \frac{\partial e_1}{\partial x_1}(x) & \dots & \frac{\partial e_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial e_m}{\partial x_1}(x) & \dots & \frac{\partial e_m}{\partial x_n}(x) \end{bmatrix}$$

- Least squares (LS) objective function

$$f(x) = \frac{1}{2} \|e(x)\|_2^2$$

$$\nabla f(x) = \left[\frac{\partial e}{\partial x}(x)\right]' e(x) = J(x)' e(x)$$

$$\nabla^2 f(x) = J(x)' J(x) + \sum_i \nabla^2 e_i(x) e_i(x) \approx J(x)' J(x)$$

- error, $e(x) = y - \hat{y}(x)$, and covariance of error, $R(x)$:

$$e(x) = \begin{bmatrix} e_1(x) \\ \vdots \\ e_m(x) \end{bmatrix}$$

$$R(x) = \begin{bmatrix} R_{11}(x) & \dots & R_{1m}(x) \\ \vdots & & \vdots \\ R_{m1}(x) & \dots & R_{mm}(x) \end{bmatrix}$$

- Maximum likelihood (ML) [negative log likelihood function]

$$f(x) = \frac{1}{2} \ln\left[\det R(x)\right] + \frac{1}{2} e(x)' R(x)^{-1} e(x)$$

$$\begin{aligned} \frac{\partial f}{\partial x_i}(x) = {} & \frac{1}{2}\mathsf{tr}\left[R(x)^{-1}\frac{\partial R}{\partial x_i}(x)\right] \\ & + e(x)'R(x)^{-1}\frac{\partial e}{\partial x_i}(x) + \frac{1}{2}e(x)'R(x)^{-1}\left[\frac{\partial R}{\partial x_i}(x)\right]R(x)^{-1}e(x) \end{aligned}$$

Parameter Estimation - Objective Functions

Regression based objective functions

- ℓ_2-regression (Least Squares, LS)

$$f(x) = \frac{1}{2}\left\|e(x)\right\|_2^2 = \frac{1}{2}\left(e_1(x)^2 + e_2(x)^2 + ... + e_N(x)^2\right)$$

- ℓ_1-regression

$$f(x) = \left\|e(x)\right\|_1 = |e_1(x)| + |e_2(x)| + \ldots + |e_N(x)|$$

- ℓ_∞-regression

$$f(x) = \left\|e(x)\right\|_\infty = \max\left\{|e_1(x)|, |e_2(x)|, \ldots, |e_N(x)|\right\}$$

- ℓ_{H_γ}-regression (Huber-regression)

$$f(x) = \left\|e(x)\right\|_{H_\gamma} = \rho_\gamma(e_1(x)) + \rho_\gamma(e_2(x)) + \ldots + \rho_\gamma(e_N(x))$$

$$\rho_\gamma(e_i(x)) = \begin{cases} \frac{1}{2}e_i(x)^2 & |e_i(x)| \leq \gamma \\ \gamma\left(|e_i(x)| - \frac{1}{2}\gamma\right) & |e_i(x)| > \gamma \end{cases}$$

Parameter Estimation - Weighted Objective Functions

Weighted errors (residuals) [scaling]

$$\varepsilon(x) = We(x)$$

Optimal scaling (given the covariance, R): $W = R^{-1/2}$

- ℓ_2-regression (Least Squares, LS)

$$f(x) = \frac{1}{2}\|We(x)\|_2^2 = \frac{1}{2}\|\varepsilon(x)\|_2^2$$

- ℓ_1-regression

$$f(x) = \|We(x)\|_1 = \|\varepsilon(x)\|_1$$

- ℓ_∞-regression

$$f(x) = \|We(x)\|_\infty = \|\varepsilon(x)\|_\infty$$

- ℓ_{H_γ}-regression (Huber-regression)

$$f(x) = \|We(x)\|_{H_\gamma} = \|\varepsilon(x)\|_{H_\gamma}$$

Parameter Estimation - ML Objective Functions

Negative log-likelihood objective function for maximum likelihood (ML) estimation

- Covariance, $R = R(x)$, unknown

$$f(x) = \frac{1}{2}\ln\left[\det R(x)\right] + \frac{1}{2}e(x)'R(x)^{-1}e(x)$$

- Covariance, R, known

$$\begin{aligned} f(x) &= \frac{1}{2}\ln\left[\det R\right] + \frac{1}{2}e(x)'R^{-1}e(x) \\ &= \frac{1}{2}\ln\left[\det R\right] + \frac{1}{2}\left\|e(x)\right\|_{R^{-1}}^2 \\ &= \frac{1}{2}\ln\left[\det R\right] + \frac{1}{2}\left\|We(x)\right\|_2^2 \qquad R^{-1} = W'W \end{aligned}$$

Therefore, we can compute the ML estimate in this case by solving the weighted LS optimization problem with the objective function

$$f(x) = \frac{1}{2}\left\|We(x)\right\|_2^2 = \frac{1}{2}\left\|\varepsilon(x)\right\|_2^2$$

where the weight matrix, $W = L^{-1}$, and L is the Cholesky factor or R, i.e. $R = LL'$, such that $R^{-1} = (L^{-1})'L^{-1} = W'W$

Parameter Estimation - ML and MAP Objective Functions

$$\min_x \; f(x)$$

Negative log likelihood functions

- Maximum Likelihood (ML)

$$f(x) = \frac{1}{2}\ln\left[\det R(x)\right] + \frac{1}{2}e(x)'R(x)^{-1}e(x)$$

- Maximum a Posteriori (MAP)

$$\begin{aligned} f(x;\theta) = &\frac{1}{2}\ln\left[\det R(x)\right] + \frac{1}{2}e(x)'R(x)^{-1}e(x) \\ &+ \frac{1}{2}\ln\left[\det P(\theta)\right] + \frac{1}{2}(x - \bar{x}(\theta))'P(\theta)^{-1}(x - \bar{x}(\theta)) \end{aligned}$$

θ is a vector of hyper-parameters that can either be fixed or part of the optimization variables, i.e.

$$\min_{x,\theta} \; f(x;\theta)$$

Parameter Estimation Algorithms

Parameter Estimation Algorithms - Gradient Based

$$\min_x \ f(x)$$

Line search:

$$\min_{p_k} \phi = \frac{1}{2} p_k' H_k p_k + \nabla f(x_k)' p_k + f(x_k)$$

$$x_{k+1} = x_k + \alpha_k p_k$$

Trust region:

$$\min_{p_k} \phi = \frac{1}{2} p_k' H_k p_k + \nabla f(x_k)' p_k + f(x_k) + \frac{1}{2}\mu_k \left\| p_k \right\|_2^2$$

$$x_{k+1} = x_k + p_k$$

- Steepest descent: $H_k = I$
  Line search: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
  Trust region: $x_{k+1} = x_k - \frac{1}{1+\mu_k} \nabla f(x_k)$
- Newton: $H_k = \nabla^2 f(x_k)$
  Line search: $x_{k+1} = x_k - \alpha_k \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$
  Trust region: $x_{k+1} = x_k - \left(\nabla^2 f(x_k) + \mu_k I\right)^{-1} \nabla f(x_k)$
- Quasi-Newton: H_k is an approximation to $\nabla^2 f(x_k)$
  Line search: $x_{k+1} = x_k - \alpha_k H_k^{-1} \nabla f(x_k)$
  Trust region: $x_{k+1} = x_k - (H_k + \mu_k I)^{-1} \nabla f(x_k)$

Parameter Estimation Algorithms - Least Squares

$$\min_x \; f(x) = \frac{1}{2}\|e(x)\|_2^2 = \frac{1}{2}e(x)'e(x), \qquad e(x) = y - \hat{y}(x)$$

- Gradient

$$\nabla f(x) = -\frac{\partial \hat{y}(x)}{\partial x}e(x) = -J(x)'e(x) \qquad J(x) = \frac{\partial \hat{y}(x)}{\partial x}$$

- Hessian

$$\nabla^2 f(x) = J(x)'J(x) - \sum_{i=1}^{N} \frac{\partial^2 \hat{y}_i(x)}{\partial x^2}e_i(x) = J(x)'J(x) + S(x)$$

  where

$$S(x) = -\sum_{i=1}^{N} \frac{\partial^2 \hat{y}_i(x)}{\partial x^2}e_i(x)$$

- Algorithms: $\nabla f(x_k) = -J(x_k)'e(x_k)$
  Line search: $x_{k+1} = x_k - \alpha_k H_k^{-1}\nabla f(x_k)$
  Trust region: $x_{k+1} = x_k - (H_k + \mu_k I)^{-1}\nabla f(x_k)$
  - Steepest descent: $H_k = I$
  - Newton: $H_k = \nabla^2 f(x_k) = J(x_k)'J(x_k) + S(x_k)$
  - Quasi-Newton: H_k an approximation to $\nabla^2 f(x_k)$
  - Gauss-Newton: $H_k = J(x_k)'J(x_k)$

Parameter Estimation Algorithm - Levenberg-Marquardt

$$\min_x f(x) = \frac{1}{2} \|e(x)\|_2^2 = \frac{1}{2} e(x)' e(x), \qquad e(x) = y - \hat{y}(x)$$

- Gradient

$$\nabla f(x) = -\frac{\partial \hat{y}(x)}{\partial x} e(x) = -J(x)' e(x) \qquad J(x) = \frac{\partial \hat{y}(x)}{\partial x}$$

- Hessian

$$\nabla^2 f(x) = J(x)' J(x) - \sum_{i=1}^{N} \frac{\partial^2 \hat{y}_i(x)}{\partial x^2} e_i(x) = J(x)' J(x) + S(x)$$

  where

$$S(x) = -\sum_{i=1}^{N} \frac{\partial^2 \hat{y}_i(x)}{\partial x^2} e_i(x)$$

- Levenberg-Marquardt Algorithm
  = Trust region algorithm with Gauss-Newton approximation:
  ($S(x_k) \approx 0$ such that $H_k = J(x_k)' J(x_k) \approx \nabla^2 f(x_k)$)

$$\begin{aligned} x_{k+1} &= x_k - (H_k + \mu_k I)^{-1} \nabla f(x_k) \\ &= x_k + \left(J(x_k)' J(x_k) + \mu_k I\right)^{-1} J(x_k)' e(x_k) \end{aligned}$$

Parameter Estimation - Basic Netwon Based Algorithm

The parameter estimation problem can be expressed as an unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

The first order (necessary but not sufficient) optimality conditions can be expressed as

$$g(x) = \nabla f(x) = 0 \qquad g : \mathbb{R}^n \mapsto \mathbb{R}^n$$

and solved using Newton's method

$$g(x_k) + \nabla g(x_k) \Delta x_k = 0$$

This is equivalent to

$$\nabla f(x_k) + \nabla^2 f(x_k) \Delta x_k = 0$$

such that

$$\Delta x_k = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$$

and

$$x_{k+1} = x_k + \Delta x_k = x_k - \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$$

Parameter Estimation

$$\min_{x} \quad f(x)$$

- Line-search based algorithm

$$x_{k+1} = x_k - \alpha_k H_k^{-1} \nabla f(x_k)$$

  - Newton: $H_k = \nabla^2 f(x_k)$
  - Steepest descent: $H_k = I$
  - Quasi-Newton: H_k is a rank-one approximation to $\nabla^2 f(x_k)$ based on gradient, $\nabla f(x_k)$, information

- Trust-region based algorithm

$$x_{k+1} = x_k - (H_k + \mu_k I)^{-1} \nabla f(x_k)$$

- These algorithms are gradient based algorithms, as they need gradient information, $\nabla f(x_k)$

Parameter Estimation

- Optimization problem

$$\min_{x} \quad f(x)$$

- Quadratic approximation

$$f(x_k + p_k) \approx f(x_k) + \nabla f(x_k)' p_k + \frac{1}{2} p_k' \nabla^2 f(x_k) p_k$$

- Quadratic program (QP) for search direction, p:

$$\min_{p_k} \quad \phi(p_k) = \frac{1}{2} p_k' H_k p_k + g_k' p_k + \rho_k$$

$$H_k = \nabla^2 f(x_k) \qquad g_k = \nabla f(x_k) \qquad \rho_k = f(x_k)$$

- Optimal solution to QP

$$\nabla \phi(p_k) = H_k p_k + g_k = 0 \qquad \Leftrightarrow \qquad p_k = -H_k^{-1} g_k$$

- Next iterate

$$x_{k+1} = x_k + \alpha_k p_k = x_k - \alpha_k H_k^{-1} g_k = x_k - \alpha_k \left[\nabla^2 f(x_k) \right]^{-1} \nabla f(x_k)$$

Parameter Estimation

- QP for search direction

$$\min_{p_k} \quad \phi(p_k) = \frac{1}{2}p_k' H_k p_k + g_k' p_k + \rho_k + \overbrace{\frac{1}{2}\mu_k \left\| p_k \right\|_2^2}^{\text{regularization term}}$$

- Objective function

$$\begin{aligned} \phi(p_k) &= \frac{1}{2}p_k' H_k p_k + g_k' p_k + \rho_k + \frac{1}{2}\mu_k \left\| p_k \right\|_2^2 \\ &= \frac{1}{2}p_k' H_k p_k + g_k' p_k + \rho_k + \frac{1}{2}\mu_k p_k' p_k \\ &= \frac{1}{2}p_k' \left(H_k + \mu_k I\right) p_k + g_k' p_k + \rho_k \end{aligned}$$

- Derivatives

$$\begin{aligned} \nabla \phi(p_k) &= \left(H_k + \mu_k I\right) p_k + g_k = 0 \\ \nabla^2 \phi(p_k) &= H_k + \mu_k I \end{aligned}$$

- Search direction / next iterate:

$$x_{k+1} = x_k + p_k = x_k - \left(H_k + \mu_k I\right)^{-1} g_k, \qquad g_k = \nabla f(x_k)$$

- Hessian approximations
  - Linear approximation / steepest descent variation: $H_k = I$
  - Newton: $H_k = \nabla^2 f(x_k)$
  - Quasi-Newton: H_k is an approximation to $\nabla^2 f(x_k)$

Parameter Estimation - Ways to create the trust region

- Regularized objective function

$$\min_{x} \quad \psi(x) = f(x) + \varphi_k(x)$$

  where e.g. $\varphi_k(x) = \mu_k \|x - x_k\|_2^2$
- Bound constrained estimation

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & l \leq x \leq u \end{aligned}$$

- Constrained estimation for the trust region

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & \|x - x_k\|_\infty \leq \Delta_k \end{aligned}$$

  is equivalent to bound constrained optimization

$$\begin{aligned} \min_{x} \quad & f(x) \\ s.t. \quad & x_k - \Delta_k e \leq x \leq x_k + \Delta_k e \end{aligned}$$

Regularization

Regularization

- Regularized optimization problem

$$\min_x \quad \psi(x) = \phi(x) + \varphi(x)$$
$$\min_x \quad \psi(x) = \phi(x) + \lambda\varphi(x)$$
$$\min_x \quad \psi(x) = \alpha\phi(x) + (1-\alpha)\varphi(x)$$

- Prediction, error and covariance

$$\hat{y} = \hat{y}(x), \qquad e(x) = y - \hat{y}(x), \qquad R = R(x)$$

- $\phi(x)$ is a function describing the fit to data

$$\phi(x) = \frac{1}{2}\left\|e(x)\right\|_2^2$$
$$\phi(x) = \frac{1}{2}\left\|W_e e(x)\right\|_2^2$$
$$\phi(x) = \frac{1}{2}\ln\left[\det R(x)\right] + \frac{1}{2}e(x)'R(x)^{-1}e(x)$$

- $\varphi(x)$ is a function describing the regularity of the solution

$$\varphi(x) = \frac{1}{2}\left\|x\right\|_2^2 \qquad \varphi(x) = \frac{1}{2}\left\|x - \bar{x}\right\|_2^2$$
$$\varphi(x) = \frac{1}{2}\left\|W_x x\right\|_2^2 \qquad \varphi(x) = \frac{1}{2}\left\|W_x(x - \bar{x})\right\|_2^2$$
$$\varphi(x) = \frac{1}{2}\ln\left[\det P\right] + \frac{1}{2}x'P^{-1}x \qquad \varphi(x) = \frac{1}{2}\ln\left[\det P\right] + \frac{1}{2}(x-\bar{x})'P^{-1}(x-\bar{x})$$

Regularization Examples

$$x = \left[x_1;\, x_2;\, \ldots;\, x_n\right], \quad x_0 = 0, \quad x_{n+1} = 0$$

- Position, x_k:

$$\varphi(x) = \frac{1}{2}\sum_{k=0}^{n+1} \|x_k\|_2^2 = \frac{1}{2}\sum_{k=1}^{n} \|x_k\|_2^2 = \frac{1}{2}\|x\|_2^2$$

- Rate, $\Delta x_k = x_k - x_{k-1}$:

$$\varphi(x) = \sum_{k=1}^{n+1} \|\Delta x_k\|_2^2 = \frac{1}{2}\|\Lambda_n x\|_2^2 \quad \Lambda_{n=4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

- Acceleration, $\Delta^2 x_k = x_{k+1} - 2x_k + x_{k-1}$

$$\varphi(x) = \frac{1}{2}\sum_{k=1}^{n} \left\|\Delta^2 x_k\right\|_2^2 = \frac{1}{2}\left\|\Lambda_n^2 x\right\|_2^2 \quad \Lambda_{n=4}^2 = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{bmatrix}$$

Regularization terms

- Quadratic regularization terms, $\varphi(x) = \frac{1}{2}x'Hx$:

$$\begin{aligned}
\varphi(x) &= \frac{1}{2}\left\|x\right\|_2^2 = \frac{1}{2}x'x \\
&= \frac{1}{2}x'Hx && H = I \\
\varphi(x) &= \frac{1}{2}\left\|W_x x\right\|_2^2 = \frac{1}{2}(W_x x)'(W_x x) = \frac{1}{2}x'\left(W_x' W_x\right)x \\
&= \frac{1}{2}x'Hx && H = W_x' W_x
\end{aligned}$$

- Linear-quadratic regularization terms, $\varphi(x) = \frac{1}{2}x'Hx + g'x + \rho$:

$$\begin{aligned}
\varphi(x) &= \frac{1}{2}\left\|x - \bar{x}\right\|_2^2 = \frac{1}{2}(x - \bar{x})'(x - \bar{x}) = \frac{1}{2}x'x - (\bar{x})'x + \frac{1}{2}\bar{x}'\bar{x} \\
&= \frac{1}{2}x'Hx + g'x + \rho, \quad H = I, \quad g = -\bar{x}, \quad \rho = \frac{1}{2}\bar{x}'\bar{x} \\
\varphi(x) &= \frac{1}{2}\left\|W_x\left(x - \bar{x}\right)\right\|_2^2 = \frac{1}{2}(W_x(x - \bar{x}))'(W_x(x - \bar{x})) \\
&= \frac{1}{2}x'\left(W_x' W_x\right)x - \left(W_x' W_x \bar{x}\right)'x + \frac{1}{2}\bar{x}'W_x' W_x \bar{x} \\
&= \frac{1}{2}x'Hx + gx + \rho, \quad H = W_x' W_x, \quad g = -W_x' W_x \bar{x}, \quad \rho = \frac{1}{2}\bar{x}'W_x' W_x \bar{x}
\end{aligned}$$

Regularization terms - gradients and Hessians

- Quadratic regularization term

$$\begin{aligned}
\varphi(x) &= \frac{1}{2}x'Hx \\
\nabla\varphi(x) &= Hx \\
\nabla^2\varphi(x) &= H
\end{aligned}$$

- Linear-quadratic regularization term

$$\begin{aligned}
\varphi(x) &= \frac{1}{2}x'Hx + g'x + \rho \\
\nabla\varphi(x) &= Hx + g \\
\nabla^2\varphi(x) &= H
\end{aligned}$$